

**COMPUTER VISION-BASED VEHICLE RECOGNITION
SYSTEM USING DEEP LEARNING TECHNIQUES**

TAN SHI HAO

**FACULTY OF ENGINEERING
UNIVERSITI MALAYA
KUALA LUMPUR**

2024

**COMPUTER VISION-BASED VEHICLE
RECOGNITION SYSTEM USING DEEP LEARNING
TECHNIQUES**

TAN SHI HAO

**THESIS SUBMITTED IN FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF DOCTOR OF
PHILOSOPHY**

**FACULTY OF ENGINEERING
UNIVERSITI MALAYA
KUALA LUMPUR**

2024

UNIVERSITI MALAYA
ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **Tan Shi Hao**

Name of Degree: **Doctor of Philosophy**

Title of Thesis: **Computer Vision-based Vehicle Recognition System Using Deep Learning Techniques**

Field of Study: **Signals & Systems, Computer Vision, Deep Learning**

I do solemnly and sincerely declare that:

- (1) I am the sole author/writer of this Work;
- (2) This Work is original;
- (3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
- (4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
- (5) I hereby assign all and every rights in the copyright to this Work to the Universiti Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
- (6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature

Date: 5/6/2024

Subscribed and solemnly declared before,

Witness's Signature

Date: 06/06/2024

Name:

Designation:

COMPUTER VISION-BASED VEHICLE RECOGNITION SYSTEM USING DEEP LEARNING TECHNIQUES

ABSTRACT

Vehicle recognition is essential for Intelligent Transportation System (ITS) in creating a comfortable commuting environment. It is the enabler for a diverse range of applications, including roadway maintenance, surveillance systems, electronic tolls, etc. With the aim of improving vehicle type and vehicle make and model recognition (VMMR) performance, the past studies are collated and a vehicle taxonomy that encompasses sensor-based and Computer Vision (CV)-based solutions is deliberated. Motivated to learn superior convolution filters, the first proposal employs Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) as filter learning techniques. The proposed network dubbed PCA-LDA-Convolutional Neural Network (CNN) also incorporates a parameter-free Channel-Based Attention Module (ChBAM) to tune the feature responses guided by the channel information saliency. The framework delivers 99.6% and 97.8% accuracies on datasets with 30 and 300 vehicle models, respectively. The robustness tests verify that PCA-LDA-CNN is steadfast against image distortions. Secondly, the past studies reveal that neglecting the degree of informativeness cripples the quality of representation learning. In this regard, a Spatial Attention Module (SAM), which is empowered by Multi-Head Self-Attention (MHSA), is proposed to scale the feature responses by exploiting spatial relevancy. The proposed ResNet50-SAM model records exceptional performance on Beijing Institute of Technology (BIT)-Vehicle, Stanford Cars and Web-Nature Comprehensive Cars (CompCarsWeb) datasets by reporting 98.2%, 84.5% and 96.0% accuracies, respectively. A qualitative inspection of the feature embeddings suggests high cohesivity within the group. Integrating SAM into other CNNs also leads to considerable improvements. Next, forgoing the low-level details and concentrating on high-level features is detrimental to VMMR. The Cross-

Granularity (CG) module, in contrast, integrates both information to render a balanced mix of local contextual information and global semantic details. The combination of ResNet50 and CG module attains 98.6%, 95.4%, 86.4% and 99.1% accuracies on CompCarsWeb, Stanford Cars, Car-FG3K and Surveillance-Nature Comprehensive Cars (CompCarsSV) datasets, respectively. The qualitative analysis further unveils its strong ability to locate the distinctive fine-grained vehicle details. The CG module is also highly compatible with various backbone CNNs. As the fourth proposal, the Coarse-to-Fine Context Aggregation (CFCA) module presents a parameter-efficient multi-scale feature learning paradigm. The cross-scale features are generated by first refining the scale-specific components independently and then fusing them in a nonlinear manner through convolution. The multi-scale feature maps produce 98.0%, 95.1%, 86.2%, 99.0%, and 96.9% accuracies on CompCarsWeb, Stanford Cars, Car-FG3K, CompCarsSV and Mohsin-VMMR datasets, respectively. Moreover, the neurons exhibit high feature responses on the discriminative vehicle parts, corresponding to the superior feature extraction ability of the CFCA module. The fifth proposal presents an Augmented-Granularity (AG) module that executes grouped focus convolution (GFConv) to compose multi-granularity features. With the spatial-to-channel transformation, the GFConv doubles the receptive field whilst mitigating information loss. When pairing the AG module with TResNet-L, the network claims 87.8%, 95.5%, 98.6% and 92.5% on Car-FG3K, Stanford Cars, CompCarsWeb and VMMRdb datasets, respectively. The dissection of the feature embeddings affirms the ability of the AG module to reduce the intraclass variance. The AG module also brings 2.7% accuracy improvements in average for 4 backbone CNNs.

Keywords: Attention, Convolutional Neural Network, Fine-Grained Visual Classification, Multi-Scale, Vehicle Recognition

SISTEM PENGENALAN KENDERAAN BERASASKAN PENGLIHATAN KOMPUTER DENGAN TEKNIK PEMBELAJARAN MENDALAM

ABSTRAK

Pengenalan kenderaan adalah penting untuk merealisasikan Sistem Kenderaan Pintar (ITS) untuk membentuk keadaan komut yang kondusif. Pengenalan kenderaan mempunyai banyak kegunaan, termasuk penyelenggaraan jalan raya, sistem pengawasan, tol elektronik dan sebagainya. Dengan objektif untuk menambah baik sistem pengenalan jenis dan model kenderaan (VMMR), kajian lepas telah diselidik dan taksonomi kenderaan yang berasaskan sensor dan penglihatan komputer (CV) telah dibentangkan. Untuk memperoleh penapis konvolusi yang berkualiti tinggi, 'Principal Component Analysis' (PCA) dan 'Linear Discriminant Analysis' (LDA) telah digunakan dalam pembentukan PCA-LDA-Rangkaian Neural Konvolusi (CNN). PCA-LDA-CNN juga mengandungi Modul Perhatian Berasaskan Saluran (ChBAM) untuk melaraskan peta ciri-ciri berpandukan kepentingan saluran. PCA-LDA-CNN masing-masing mencapai ketepatan 99.6% dan 97.8% dalam membezakan 30 dan 300 model kenderaan. Selain itu, gambar yang berkualiti rendah mempunyai pengaruh yang terhad terhadap ketepatan PCA-LDA-CNN. Kajian lepas mendedahkan bahawa pengabaian tahap infomasi menjejaskan kualiti pembelajaran model. Bagi menangani masalah ini, 'Spatial Attention Module' (SAM) yang memperalatkan 'Multi-Head Self-Attention' (MHSA) dicadangkan untuk melaraskan peta ciri-ciri dengan mengeksploitasi perkaitan spatial. Cadangan model ResNet50-SAM mencapai keputusan klasifikasi yang mengagumkan dalam data Beijing Institute of Technology (BIT)-Vehicle, Stanford Cars dan Web-Nature Comprehensive Cars (CompCarsWeb) dengan ketepatan masing-masing 98.2%, 84.5% dan 96.0%. Pemeriksaan ciri-ciri kenderaan model yang telah dipelajari menonjolkan kohesi yang tinggi. Ketepatan CNN yang lain juga telah ditingkatkan selepas digabungkan dengan SAM. Pengabaian butiran bertahap rendah dan hanya menumpukan

perhatian pada ciri-ciri bertahap tinggi memudaratkan klasifikasi model kenderaan. Sebagai jalan penyelesaian, modul ‘Cross-Granularity’ (CG) telah dicipta untuk menjamin campuran yang seimbang antara informasi kontekstual tempatan dan butiran semantik global. Kombinasi ResNet50 dan modul CG masing-masing mencecah ketepatan 98.6%, 95.4%, 86.4 dan 99.1% dalam data CompCarsWeb, Stanford Cars, Car-FG3K dan Surveillance-Nature Comprehensive Cars (CompCarsSV). Di samping itu, analisis menunjukkan keupayaan modul CG dalam mengenal pasti bahagian-bahagian kenderaan yang unik. Modul CG juga dibuktikan memiliki keserasian yang tinggi dengan CNN yang lain. Modul ‘Coarse-to-Fine Context Aggregation’ (CFCA) mencadangkan kaedah pembelajaran ciri-ciri multi-skala yang cekap. Ciri-ciri tersebut dihasilkan dengan memperhalusi komponen yang berasal dari berbagai-bagai skala secara berasingan sebelum mencampurkan mereka melalui konvolusi. Peta ciri-ciri multi-skala ini adalah sangat berkesan dengan masing-masing merekodkan ketepatan 98.0%, 95.1%, 86.2%, 99.0%, dan 96.9% dalam data CompCarsWeb, Stanford Cars, Car-FG3K, CompCarsSV dan Mohsin-VMMR. Tambahan pula, neuron-neuron mempunyai kadar sensitiviti yang tinggi terhadap bahagian kenderaan yang ketara dan ini membuktikan keunggulan modul CFCA dalam process pembelajaran. Modul ‘Augmented-Granularity’ (AG) menggunakan ‘grouped focus convolution’ (GFConv) untuk menghasilkan ciri-ciri multi-skala. Dengan menjalani transformasi ruang ke saluran, GFConv berjaya untuk menambah medan penerimaan dua kali ganda dan mengurangkan kehilangan informasi. Apabila digunakan dengan TResNet-L, ketepatan masing-masing 87.8%, 95.5%, 98.6% dan 92.5% telah dicecah untuk data Car-FG3K, Stanford Cars, CompCarsWeb dan VMMRdb. Pemeriksaan ciri-ciri yang telah dipelajari mendedahkan bahawa keupayaan modul AG dalam mengurangkan varians dalam kelas. Modul AG juga meningkatkan ketepatan untuk 4 CNN yang lain dengan purate 2.7%.

Kata kunci: Perhatian, Rangkaian Neural Konvolusi, Klasifikasi Visual Berbutir Halus, Pelbagai skala, Pengenalan Kenderaan

Universiti Malaya

ACKNOWLEDGEMENTS

I would like to extend my heartfelt appreciation to my supervisor, Prof. Ir. Ts. Dr. Chuah Joon Huang for his unconditional support and constructive advice throughout the PhD journey. His professional guidance enlightens me to steer the research in the right direction and empowers me to come up with solutions swiftly in the face of stumbling blocks. He also invests immense energy and time in reviewing the manuscripts and thesis to safeguard the quality of the deliverables. Without his ongoing support, the journey would be excruciating. The co-supervisors, who are Assoc. Prof. Ir. Dr. Chow Chee Onn and Assoc. Prof. Ir. Dr. Jeevan A/L Kanesan, are also highly appreciated for having full confidence in me to complete the study.

Furthermore, I am deeply grateful for having my grandparents, Mr. Tan Hwa Hong and Mdm Yau Siew Eng, parents, Mr. Tan Kok Huat and Mdm Sia Fong Choo, as well as my two beloved sisters, Ms. Tan Jia Ying and Ms Tan Ke Ying, standing by my side and offering me words of encouragement. Those wise words are invaluable and have energized me to endure all the challenges throughout my PhD journey.

Most important of all, I would like to express my utmost gratitude to Ms. Tan Mei Pheng for showering me with selfless love and providing me the relentless support to accomplish what I have set out to do. Her companion has consistently blessed me with unwavering perseverance and made the PhD study more exhilarating than ever. Her piece of advice on the produced works has also been one of the contributing factors in improving the fineness of final deliverables.

TABLE OF CONTENTS

Abstract	iii
Abstrak	v
Acknowledgements	viii
Table of Contents	ix
List of Figures	xv
List of Tables.....	xviii
List of Symbols and Abbreviations.....	xxi
CHAPTER 1: INTRODUCTION.....	1
1.1 Overview.....	1
1.2 Problem Statement.....	5
1.3 Objectives	7
1.4 Scope of Research.....	8
1.5 Significance of Research	10
1.6 Thesis Outline.....	11
CHAPTER 2: LITERATURE REVIEW.....	14
2.1 Introduction.....	14
2.2 Machine Learning	15
2.3 Deep Learning	16
2.3.1 Recurrent Neural Network	20
2.3.2 Convolutional Neural Network	21
2.3.3 Transformer	23
2.3.4 Fully Connected Layer	23
2.3.5 Loss Function	23

2.3.6	Backpropagation.....	26
2.3.7	Optimization.....	26
2.3.8	Pretrained Convolutional Neural Network.....	27
2.3.8.1	AlexNet	28
2.3.8.2	VGG.....	29
2.3.8.3	Inception Network.....	29
2.3.8.4	Residual Network.....	30
2.3.8.5	Densely Connected Convolutional Network.....	30
2.3.8.6	Efficient Network.....	30
2.4	Vehicle Recognition System.....	31
2.4.1	Sensor	33
2.4.1.1	On-Roadway.....	33
2.4.1.2	Side-Roadway	37
2.4.1.3	Over-Roadway	40
2.4.2	Computer Vision	42
2.4.2.1	Model-Based	42
2.4.2.2	Feature-Based.....	46
2.4.2.3	Novel Backbones.....	51
2.4.2.4	Unsupervised Filter Learning.....	60
2.4.2.5	Part-Based	64
2.4.2.6	Attention.....	74
2.4.2.7	Multi-Scale Features	81
2.5	Summary.....	88

CHAPTER 3: PCA-LDA-BASED CONVOLUTIONAL NEURAL NETWORK WITH CHANNEL-BASED ATTENTION MODULE FOR VEHICLE MAKE AND MODEL RECOGNITION	90
---	-----------

3.1	Introduction.....	90
3.2	Literature Review	91
3.3	Methodology.....	93
3.3.1	Feature Extractor	93
3.3.2	PCA-LDA-CNN.....	94
3.3.2.1	Generation of PCA Filters.....	95
3.3.2.2	Generation of LDA Filters	96
3.3.2.3	Channel-Based Attention Module.....	97
3.4	Experiments	98
3.4.1	Datasets	98
3.4.2	Implementation Details	99
3.5	Results & Discussions	100
3.5.1	Quantitative Analysis	100
3.5.2	Ablation Study.....	103
3.5.3	Robustness Test.....	104
3.5.3.1	Translation.....	104
3.5.3.2	Scaling.....	105
3.5.3.3	Rotation	106
3.5.3.4	Brightness and Contrast	107
3.6	Conclusion	108

CHAPTER 4: SPATIALLY RECALIBRATED CONVOLUTIONAL NEURAL NETWORK FOR VEHICLE TYPE RECOGNITION 109

4.1	Introduction.....	109
4.2	Literature Review	110
4.3	Methodology.....	114
4.3.1	Spatial Attention Module	114

4.3.1.1	Preprocessing of Feature Maps	115
4.3.1.2	Injection of Positional Information	116
4.3.1.3	Multi-Head Self-Attention	116
4.3.1.4	Recalibration of Feature Maps	117
4.3.2	Spatially Recalibrated Convolutional Neural Network.....	118
4.4	Experiments	119
4.4.1	Datasets	119
4.4.2	Implementation Details	121
4.5	Results & Discussions	121
4.5.1	Quantitative Analysis	121
4.5.2	Qualitative Analysis	127
4.5.3	Ablation Study.....	127
4.5.3.1	Effect of Positional Information On SAM	128
4.5.3.2	Effect of Patch Size on SAM	128
4.5.3.3	Effect of Number of Heads on SAM.....	129
4.5.4	Generalization Study	130
4.5.5	Performance of SAM against Existing Attention Modules.....	131
4.6	Conclusion.....	132

**CHAPTER 5: CROSS-GRANULARITY NETWORK FOR VEHICLE MAKE
AND MODEL RECOGNITION..... 133**

5.1	Introduction.....	133
5.2	Literature Review	134
5.3	Methodology.....	137
5.3.1	Cross-Granularity Module.....	137
5.3.2	Cross-Granularity Network	140
5.3.3	Loss Function	141

5.4	Experiments	142
5.4.1	Datasets	142
5.4.2	Implementation Details	143
5.5	Results & Discussions	144
5.5.1	Quantitative Analysis	144
5.5.2	Qualitative Analysis	150
5.5.3	Ablation Study.....	153
5.5.4	Generalization Study	155
5.6	Conclusion	155
CHAPTER 6: COARSE-TO-FINE CONTEXT AGGREGATION NETWORK FOR VEHICLE MAKE AND MODEL RECOGNITION		157
6.1	Introduction.....	157
6.2	Literature Review	158
6.3	Methodology.....	161
6.3.1	Coarse-to-Fine Context Aggregation Module	162
6.3.2	Coarse-to-Fine Context Aggregation Network	165
6.3.3	Loss Function	167
6.4	Experiments	167
6.4.1	Datasets	167
6.4.2	Implementation Details	169
6.5	Results & Discussions	170
6.5.1	Quantitative Analysis	170
6.5.2	Qualitative Analysis	179
6.5.3	Ablation Study.....	183
6.5.4	Generalization Study	185
6.6	Conclusion	187

CHAPTER 7: AUGMENTED-GRANULARITY NETWORK FOR VEHICLE MAKE AND MODEL RECOGNITION	188
7.1 Introduction.....	188
7.2 Literature Review	188
7.3 Methodology.....	192
7.3.1 Augmented-Granularity Module	192
7.3.2 Augmented-Granularity Network.....	197
7.3.3 Loss Function	198
7.4 Experiments	199
7.4.1 Datasets	199
7.4.2 Implementation Details	201
7.5 Results & Discussions	202
7.5.1 Quantitative Analysis	202
7.5.2 Qualitative Analysis	207
7.5.3 Ablation Study.....	208
7.5.4 Generalization Study	210
7.6 Conclusion.....	212
CHAPTER 8: CONCLUSION.....	213
8.1 Conclusion	213
8.2 Future Works	215
References.....	218
List of Publications and Papers Presented	245

LIST OF FIGURES

Figure 1.1: Illustration of Vehicle Type, Vehicle Logo, Vehicle Make and Vehicle Make and Model Recognition.....	4
Figure 2.1: Artificial Intelligence, Machine Learning and Deep Learning.....	15
Figure 2.2: Biological Neuron.....	17
Figure 2.3: Perceptron.....	17
Figure 2.4: Multi-Layer Perceptron	19
Figure 2.5: Long Short-Term Memory and Gated Recurrent Unit	20
Figure 2.6: Convolutional Neural Network	21
Figure 2.7: AlexNet.....	28
Figure 2.8: Inceptionv4.....	29
Figure 2.9: Taxonomy of Vehicle Recognition System.....	31
Figure 2.10: Operating Principle of Magnetic Sensor	34
Figure 2.11: Experiment Setup for LiDAR-based VTR	37
Figure 2.12: A Hybrid System for VTR	40
Figure 2.13: Example of Original Vehicles (Top), 3D Bounding Box (Middle) and Unpacked Version (Bottom)	44
Figure 2.14: Bag of Expressions	48
Figure 2.15: Fine-Grained Transformer.....	56
Figure 2.16: Local Tiled CNN.....	61
Figure 2.17: End-to-end Localization and Classification Model.....	68
Figure 2.18: ML-Decoder	77
Figure 2.19: Cross-Part CNN.....	83
Figure 3.1: CaffeNet.....	93
Figure 3.2: PCA-LDA-CNN	95

Figure 3.3: Channel-Based Attention Module	97
Figure 3.4: Samples Images from BVMMRv2 and CompCarsSV	98
Figure 3.5: Images Translated with (a) -10, -10, (b) -10, 10, (c) 10, -10, (d) 10, 10	104
Figure 3.6: Images with Scaling Factor of (a) 0.6 (b) 0.8 (c) 1.2 (d) 1.4.....	105
Figure 3.7: Robustness Test Against Scaling.....	106
Figure 3.8: Images with Rotation (a) -20° (b) -10° (c) 10° (d) 20°	106
Figure 3.9: Robustness Test Against Rotation.....	107
Figure 3.10: Images adjusted with λ and η of (a) 0.5, -20 (b) 0.5, 20 (c) 2, -20 (d) 2, 20	107
Figure 4.1: Spatial Attention Module	115
Figure 4.2: CaffeNet.....	119
Figure 4.3: Sample Images from BIT-Vehicle Dataset.....	120
Figure 4.4: TSNE Plot of Deep Features from CaffeNet-SAM.....	127
Figure 4.5: Effect of Number of Heads on CaffeNet-SAM.....	130
Figure 4.6: Compatibility between the SAM and Existing CNNs on BIT Vehicle (Subset)	130
Figure 5.1: Cross-Granularity Network	138
Figure 5.2: Sample Images from CompCarsWeb, Stanford Cars, Car-FG3K and CompCarsSV	142
Figure 5.3: Compatibility between CG Module and Existing CNNs on CompCarsWeb	155
Figure 6.1: Coarse-to-Fine Context Aggregation Network.....	162
Figure 6.2: Sample Images from CompCarsWeb, Stanford Cars, Car-FG3K, CompCarsSV and Mohsin-VMMR	169
Figure 6.3: Confusion Matrix of CFCANet for VMMR-Mohsin	179
Figure 6.4: Suzuki Carry (Top Row) and Suzuki Highroof (Bottom Row).....	179
Figure 6.5: Visualization of Feature Embeddings.....	181

Figure 6.6: Compatibility between CFCA module and Existing CNNs on CompCarsWeb	186
Figure 6.7: Compatibility between the CFCA module and Existing CNNs on Stanford Cars	186
Figure 7.1: Augmented-Granularity Network.....	193
Figure 7.2: Sample Images from Car-FG3K, Stanford Cars, CompCarsWeb and VMRRdb.....	201
Figure 7.3: Visualization of Feature Embeddings.....	208
Figure 7.4: Compatibility between AG Module and Existing CNNs on Car-FG3K	211

Universiti Malaysia

LIST OF TABLES

Table 2.1: Summary of On-Roadway Methods	36
Table 2.2: Summary of Side-Roadway Methods	39
Table 2.3: Summary of Over-Roadway Methods	41
Table 2.4: Summary of Model-Based Methods	45
Table 2.5: Summary of Feature-Based Methods.....	50
Table 2.6: Summary of Novel Backbones	58
Table 2.7: Summary of Unsupervised Filter Learning	63
Table 2.8: Summary of Part-Based Methods	72
Table 2.9: Summary of Attention-Based Methods	79
Table 2.10: Summary of Multi-Scale Features-Based Methods	86
Table 3.1: Breakdown of Daytime and Night-time Images	99
Table 3.2: Training Duration of Proposed Framework.....	100
Table 3.3: Performance Benchmarking on Primary Dataset.....	101
Table 3.4: Performance Breakdown Analysis on Primary Dataset.....	102
Table 3.5: Performance Benchmarking on Secondary Dataset.....	103
Table 3.6: Ablation Study	103
Table 3.7: Robustness Test against Translation.....	105
Table 3.8: Robustness Test against Lighting and Contrast.....	108
Table 4.1: Statistics of Stanford Cars and CompCarsWeb	120
Table 4.2: Performance Benchmarking on BIT-Vehicle (Subset)	121
Table 4.3: Performance Benchmarking on BIT-Vehicle (Full)	124
Table 4.4: Performance Benchmarking on Stanford Cars and CompCarsWeb	126
Table 4.5: Computational Complexity of SAM.....	126

Table 4.6: Effect of Positional Information on SAM.....	128
Table 4.7: Effect of Patch Size on CaffeNet-SAM.....	129
Table 4.8: Performance Comparisons against Existing Attention Modules on BIT-Vehicle (Subset)	132
Table 5.1: Configuration of Classification Head	140
Table 5.2: Datasets Statistics	143
Table 5.3: Performance Benchmarking on CompCarsWeb.....	145
Table 5.4: Performance Benchmarking on Stanford Cars	146
Table 5.5: Performance Benchmarking on Car-FG3K	148
Table 5.6: Performance Benchmarking on CompCarsSV	150
Table 5.7: Visualization of Feature Maps Using Grad-CAM.....	152
Table 5.8: Ablation Study on CompCarsWeb	153
Table 6.1: Datasets Statistics	169
Table 6.2: Performance Benchmarking on CompCarsWeb.....	171
Table 6.3: Performance Benchmarking on Stanford Cars	173
Table 6.4: Performance Benchmarking on Car-FG3K	175
Table 6.5: Performance Benchmarking on CompCarsSV	177
Table 6.6: Performance Benchmarking on Mohsin-VMMR	177
Table 6.7: Performance Breakdown Analysis of CFCANet on Mohsin-VMMR.....	178
Table 6.8: Visualization of Feature Maps Using Grad-CAM.....	182
Table 6.9: Ablation Study on CompCarsWeb	183
Table 6.10: Ablation Study on Stanford Cars	183
Table 7.1: Datasets Statistics	201
Table 7.2: Performance Benchmarking on Car-FG3K	202
Table 7.3: Performance Benchmarking on Stanford Cars	205

Table 7.4: Performance Benchmarking on CompCarsWeb	207
Table 7.5: Performance Benchmarking on VMRRdb	207
Table 7.6: Ablation Study on Car-FG3K	210
Table 7.7: Ablation Study on Stanford Cars	210

Universiti Malaya

LIST OF SYMBOLS AND ABBREVIATIONS

X_i	:	i^{th} Input Neuron
W_i	:	Weights for X_i
b	:	Bias
Y	:	Output Response produced by Perceptron
Z	:	Y after applying the Activation Function
I_i	:	i^{th} Input Image
I_{conv}	:	Feature Maps Post-Convolution Operation
I_{pool}	:	Feature Maps Post-Pooling Operation
L	:	Number of Classes
l	:	l^{th} Class out of L Number of Classes
x_l	:	l^{th} output neuron from the fully connected layer
N_{Train}	:	Number of Training Images
$y_{i,l}$:	Binary Indicator of Class l for i^{th} Input Image
$p_{i,l}$:	Predicted Probability of Class l for i^{th} Input Image
ϵ	:	Label Smoothing Coefficient
$z_{emb,i}$:	Deep Feature Embeddings for I_i
ω	:	Margin used in Contrastive Loss
γ	:	Focal Factor
θ	:	Trainable Parameters
β	:	Exponential Decay
V_t	:	Gradient at Iteration t
β_1	:	Exponential Decay Rate for First Moment Estimate
β_2	:	Exponential Decay Rate for Second Moment Estimate
m_t	:	First Moment Estimate before Bias Correction

\hat{m}_t	:	First Moment Estimate after Bias Correction
v_t	:	Second Moment Estimate before Bias Correction
\hat{v}_t	:	Second Moment Estimate after Bias Correction
ϵ	:	Epsilon
a_i	:	A Single Feature Response from i^{th} Channel
$a_{i,norm}$:	A Normalized Feature Response from i^{th} Channel
C	:	Number of Channels
H	:	Height
W	:	Width
α_{LRN}	:	Alpha for Local Response Normalization Layer
β_{LRN}	:	Beta for Local Response Normalization Layer
r	:	Radius for Local Response Normalization Layer
k	:	Kernel Size
P	:	Number of Patches
$I_{i,p}^{patch}$:	p^{th} Image Patch for I_i
I_i^{patch}	:	Image Patches of I_i
N_{PCA}	:	Number of Required Principal Component Analysis Filters
G	:	Eigenvectors of Principal Component Analysis
I	:	Identity Matrix
S_l	:	Set of Indices for the Inputs under Class l
μ_l	:	Class Mean
$S_{w,l}$:	Within-Class Scatter Matrix for Class l
μ	:	Overall Mean
S_b	:	Between-Class Scatter Matrix
N_{LDA}	:	Number of Required Linear Discriminant Analysis Filters

E	:	Eigenvectors of Linear Discriminant Analysis
z_c	:	Maximum Feature Response from c^{th} Channel
TP	:	True Prediction Count
N_{Test}	:	Number of Testing Images
Λ	:	Parameter for Contrast Adjustment
η	:	Parameter for Brightness Adjustment
I'	:	Image after Contrast and Brightness Adjustment
s	:	Layer Index
I_i^s	:	Feature maps of I_i at Layer s
I_i^{proj}	:	Linearly Projected I_i^{patch}
W_{proj}	:	Matrices of Linear Projection
P_{enc}	:	Positional Encoding
I_i^{pos}	:	Addition of I_i^{proj} and P_{enc}
Q	:	Query Matrix of Multi-Head Self-Attention
K	:	Key Matrix of Multi-Head Self-Attention
V	:	Value Matrix of Multi-Head Self-Attention
Hd	:	Number of Heads
$Head_i$:	i^{th} Head of Multi-Head Self-Attention
d_i	:	Embedding Dimension of $Head_i$
W^O	:	Final Linear Projection Matrix of Spatial Attention Module
I_i^{MHA}	:	Output Feature Maps from Spatial Attention Module
$I_i^{MHA'}$:	I_i^{MHA} after Reshape Operation
$I_i^{S'}$:	I_i^S after Multiplication with $I_i^{MHA'}$
TP_l	:	True Prediction for Class l
N_l	:	Total Image Count for Class l

N_{CB}	:	Number of Convolutional Blocks
$I_{CB,i}$:	Feature Maps produced by i^{th} Convolutional Block
C_{FE}	:	Number of Output Channels by Feature Extraction Stage
$I_{DS,i}$:	Feature Maps after Depth Standardization Operation on $I_{CB,i}$
d	:	Dilation Rate
$I_{SR,i}$:	Feature Maps after Spatial Reduction Operation on $I_{DS,i}$
I_{FI}	:	Feature Maps produced by the Feature Integration Stage
C_{FI}	:	Channel Count of I_{FI}
FC_L	:	Fully Connected Layer with L Output Neurons
μ_{cls}	:	Mean Image Count per Class
σ_{cls}	:	Standard Deviation of Image Count per Class
I_{stem}	:	Feature Maps produced by Stem Unit
N_{CFCA}	:	Number of Coarse-to-Fine Context Aggregation Modules
$I_{low,i}$:	Low-Level Feature Maps of Coarse-to-Fine Context Aggregation Modules
$I_{high,i}$:	High-Level Feature Maps of Coarse-to-Fine Context Aggregation Modules
N_d	:	Maximum Dilation Rate in Use
$I'_{low,i}$:	Low-Level Feature Maps After Feature Extraction Stage
$I'_{high,i}$:	High-Level Feature Maps After Feature Extraction Stage
g	:	Number of Groups for Grouped Convolution
N_{FE}	:	Number of Feature Pyramids to Use
C^t	:	Class Feature Center Matrix at Iteration t
$LOSS_{LS}$:	Label Smoothing Loss
$LOSS_{center}$:	Center Loss

α_{center}	:	Contribution of Center Loss
A3M	:	Attribute-Aware Attention Model
Adam	:	Adaptive Moment Estimation
AG	:	Augmented-Granularity
AffNet	:	Affine Network
AI	:	Artificial Intelligence
AOLM	:	Attention Object Location Module
APCNN	:	Attention Pyramid CNN
APINet	:	Attentive Pairwise Interaction Network
AttNet	:	Attention Network
BAM	:	Bottleneck Attention Module
BB	:	BubbleBank
BIT	:	Beijing Institute of Technology
BN	:	Batch Normalization
BoE	:	Bag of Expressions
BoF	:	Bag of Features
BoSURF	:	Bag of Speeded-Up Robust Features
BP	:	Bilinear Pooling
BRISK	:	Binary Robust Invariant Scalable Keypoints
C3S	:	Cross-Category Cross-Semantic
CA	:	Cross Attention
CA-MSNet	:	Multi-Scale Sparse Network with Cross-Attention mechanism
CAD	:	Computer-Aided-Design
CAiT	:	Class-Attention in Image Transformers
CAL	:	Counterfactual Attention Learning
CAM	:	Class Activation Mapping

CAP	:	Context-aware Attentional Pooling
CB	:	Convolutional Block
CBAM	:	Convolutional Block Attention Module
CCI	:	Contrastive Channel Interaction
CeiT	:	Convolution-enhanced image Transformer
CFCA	:	Coarse-to-Fine Context Aggregation
ChBAM	:	Channel-Based Attention Module
CIN	:	Channel Interaction Network
CNN	:	Convolutional Neural Network
CG	:	Cross-Granularity
CL	:	Cross-Layer
CLANet	:	Cross-Layer Attention Network
CLCA	:	Cross-Layer Context Attention
CLSA	:	Cross-Layer Spatial Attention
CMAL	:	Cross-Layer Mutual Attention Learning
CMP	:	Channel-wise Max Pooling
CompCarsWeb	:	Web-Nature Comprehensive Cars
CompCarsSV	:	Surveillance-Nature Comprehensive Cars
ConvAM	:	Convolutional Attention Model
CP-CNN	:	Cross-Part CNN
CRA-CNN	:	Contrastively-Reinforced Attention CNN
CSI	:	Channel State Information
CT	:	Context Transformer
CV	:	Computer Vision
DADAINet	:	Data Augmented Dual-Attention Interactive Network
DANet	:	Dense Attention Network

DARTS	:	Differential Architecture Search
DenseNet	:	Densely Connected Convolutional Network
DFL	:	Distractive Feature Learning
DFT	:	Discrete Fourier Transform
DL	:	Deep Learning
DN	:	Deep Network
DPM	:	Deformable Part Models
DSRC	:	Discriminative Sparse Representation-based Classification
DWT	:	Discrete Wavelet Transform
EfficientNet	:	Efficient Network
ELoPE	:	Efficient Localization, Pooling and Embedding Network
EPCNN	:	Embedding Pose CNN
FE	:	Feature Extraction
FEB	:	Feature Enhancement Block
FF-CMNet	:	Feature Fusion Car Model Classification Network
FFAc	:	Feature in Feature Abstraction
FFN	:	Feed Forward Network
FI	:	Feature Integration
FIFFNet	:	Feature Integration and Feature Fusion Network
FLOPs	:	Floating Point Operations
FPN	:	Feature Pyramid Network
FPS	:	Frame per Second
FRenNet	:	Feature Relocation Network
FSRA	:	Feature Spatial Relationship Attention
GAC-CNN	:	Global Attention-Concentrated CNN
GAP	:	Global Average Pooling

GCN	:	Graph Convolutional Network
GDSMPNet	:	Granularity-aware Distillation and Structure Modeling Region Proposal Network
GeLU	:	Gaussian Error Linear Unit
GFConv	:	Grouped Focus Convolution Layer
GFMA	:	Global Feature Map Attention
GFN	:	Global Filter Network
GIAN	:	Global Information-Assisted Network
GNN	:	Graph Neural Network
GPU	:	Graphic Processing Unit
Grad-CAM	:	Gradient-Weight-Class Activation Mapping
GRU	:	Gated Recurrent Unit
GTCNet	:	Global Topology Constraint Network
GWT	:	Gabor Wavelet Transform
HERBS	:	High-temperature Refinement and Background Suppression
HOG	:	Histogram of Oriented Gradient
HSFA	:	Hierarchy Stage Feature Aggregation
I2T	:	Image-to-Tokens
IRVD	:	Iranian Vehicle Dataset
ITS	:	Intelligent Transportation System
JF	:	Joint Fine-tuning
KD	:	Knowledge Distillation
KL	:	Kullback-Leibler
kNN	:	k-Nearest Neighbours
kNNPC	:	kNN Probability Classifier
LBP	:	Local Binary Pattern

LDA	:	Linear Discriminant Analysis
LEFFN	:	Locally-Enhanced FFN
LGBPHS	:	Local Gabor Binary Pattern Histogram Sequence
LiDAR	:	Light Detection and Ranging
LLC	:	Locality-constraint Linear Coding
LNHS	:	Locally Normalized Harris Strength
LPR	:	License Plate Recognition
LRAU	:	Lightweight RAU
LRN	:	Local Response Normalization
LSTM	:	Long Short-Term Memory
LTCNN	:	Local Tiled CNN
LWCNN	:	Lightweight CNN
LWCTA	:	Layer-Wise Class Token Attention
MA-CNN	:	Multi-Attention CNN
MA-Recall	:	Macro Average Recall
MAS	:	Multi-Agent Systems
MAWNet	:	Multiscale Attention Windows Network
MD	:	Modular-Dictionary
ME-ASNet	:	Mixture-of-Expert-Attention Swapping Network
MFF	:	Multilayers Feature Fusion
MGFL	:	Multi-Granularity Feature Learning
MHCA	:	Multi-Head Class Attention
MHSA	:	Multi-Head Self-Attention
MIO-TCD	:	Miovision Traffic Camera Dataset
ML	:	Machine Learning
MLBPNNet	:	Multilayer Bilinear Pooling Network

MLP	:	Multi-Layer Perceptron
MMALNet	:	Multi-Branch and Multi-Scale Learning Networks
MS-DRAN	:	Multi-Scale Discriminative Regions Attention Network
MSN	:	Multi-Stream Network
MSPyConv	:	Multi-Scale Pyramid Convolution
MSS	:	Multi-Scale Sparse
MultiSE	:	Multi Squeeze-Excitation
NAS	:	Neural Architecture Search
NLP	:	Natural Language Processing
NMS	:	Non-Maximal Suppression
NN	:	Nearest Neighbours
NTOU-MMR	:	National Taiwan Ocean University-Make and Model Recognition
OSME	:	One-Squeeze Multi-Excitation
PAN	:	Path Aggregation Network
PCA	:	Principal Component Analysis
PCB	:	Parallel Convolutional Block
PHOG	:	Pyramid Histogram of Oriented Gradient
PLFENet	:	Part-Level Feature Extraction Network
PMG	:	Progressive Multi-Granularity
PSDPNet	:	Progressively Sampling Discriminative Parts Network
PSM	:	Part Selection Module
PyCB	:	Pyramid Context Block
PyConv	:	Pyramid Convolution
RA-CNN	:	Recurrent Attention-CNN
RADAR	:	Radio Detection and Ranging

RAU	:	Recurrent Attention Unit
RBF	:	Radial Basis Function
ReLU	:	Rectified Linear Unit
ResNet	:	Residual Network
RF	:	Radio Frequency
RHLFL	:	Relocated High-Level Feature Learning
RNN	:	Recurrent Neural Network
ROI	:	Region of Interest
RPN	:	Region Proposal Network
RSSI	:	Received Signal Strength Indicator
SA-MFNetc	:	Self-supervised Attention Filtering and Multi-Scale Features Network
SAM	:	Spatial Attention Module
SCI	:	Self-Channel Interaction
SE	:	Squeeze and Excitation
SF	:	Sparse Filtering
SGD	:	Stochastic Gradient Descent
SIFT	:	Scale Invariant Feature Transform
SLFL	:	Sparse Laplacian Filter Learning
SPM	:	Spatial Pyramid Matching
SSD	:	Single Shot Detector
SSLNet	:	Siamese Self-Supervised Learning Network
STN	:	Spatial Transformer Network
STSC	:	Spatiotemporal Sample Consistency
SURF	:	Speeded Up Robust Feature
SUV	:	Sport Utility Vehicles

SVM	:	Support Vector Machine
SWP	:	Spatially Weighted Pooling
T-S-CNN	:	Teacher-Student-Based Attention CNN
TICA	:	Topographic Independent Component Analysis
TransFG	:	Fine-Grained Transformer
TransIFC	:	Invariant Cues-Aware Feature Concentration Transformer
TransIFC+	:	Invariant cues-aware Feature Concentration Transformer Plus
TSNE	:	T-distributed Stochastic Neighbor Embedding
ViT	:	Vision Transformer
VLD	:	Vehicle Logo Detection
VLR	:	Vehicle Logo Recognition
VMMR	:	Vehicle Make and Model Recognition
VMR	:	Vehicle Make Recognition
VTID	:	Vehicle Type Image Data
VTR	:	Vehicle Type Recognition
WDNN	:	Wavelet Deep Neural Network
WS-DAN	:	Weakly-Supervised Data Augmentation Network
YOLO	:	You Only Look Once
YOLOR	:	You Only Learn One Representation

CHAPTER 1: INTRODUCTION

1.1 Overview

Advancements in technology have eased human mobility to an unprecedented extent. Ground transportation such as motorcycles, cars, lorries, subways, etc., revolutionizes the traveling means and there has been a tremendous increase in traffic volume given the growing human population on Earth. To allow better traffic regulation, a concept called Intelligent Transportation System (ITS) is introduced. The vision is to efficiently monitor and regulate the traffic as well as to help in the planning of travel routes by leveraging the data. This will eventually render a seamless traveling experience through the creation of a safe and reliable traveling environment.

One of the most notable applications of ITS is the autopilot function by Tesla (Ingle & Phute, 2016), Mercedes-Benz Group AG (Hermann, 2018) and Google (Baruah et al., 2019; Jeng et al., 2013). Such functionality allows the vehicle to be maneuvered with little to no intervention from humans. Other applications of ITS include vehicle counting, vehicle reidentification, vehicle tracking, overspeeding detection, percentage of lane usage, etc (Arinaldi et al., 2018; Kul et al., 2017; Wen et al., 2015; Won, 2020).

Nevertheless, the precursor to the successful implementation of the aforementioned applications is vehicle recognition. Vehicle recognition is defined as a process of categorizing vehicles into their respective groups (Otto, 2006; Sotheany & Nuthong, 2017; Sun, 2000; Sun & Ritchie, 2000; Tripathi et al., 2015; Yao et al., 2016). Acquiring the amount and the types of vehicles on the road can provide an invaluable reference for the assessment of road health conditions (Shokravi & Bakhary, 2017; Shokravi et al., 2020a, 2020c; Shokravi et al., 2020d, 2020e) and thereby allows the authority to make an informed decision in scheduling pavement maintenance work. Additionally, this information is worth being considered during the geometric roadway design. For instance,

the design of roadways should avoid sharp corners and narrow vehicle lanes if long trailer trucks are mostly seen on the road. Apart from that, security enforcement is another potential application (Tamam et al., 2020). An alarm can be immediately raised to security personnel if any blacklisted vehicle types or vehicle models appear within the camera view. Criminal case investigation would also be beneficial where the investigation officers can make use of the vehicle recognition system to identify the suspicious vehicle models and subsequently perform vehicle trajectory analysis (Xie et al., 2021) without the need for manual screening of the video footage. Moreover, another practical application is the automation of toll fare collection (Sun et al., 2017; Zhu & Guo, 2012). The toll fare is charged based on the vehicle types and the human workforce is usually deployed to perform the recognition. By installing the vehicle recognition and payment system, the toll fare collection process can be done swiftly. Vehicle model information is also helpful to the formulation of targeted fuel subsidies by policymakers. Those who drive posh cars can be regarded as an affluent group and more fuel subsidies should be channeled to those with low-to-middle-range cars.

Given the immense number of applications, there is a pressing need for an accurate vehicle recognition solution. In the early times, intelligent-minded human was deployed to perform the task. However, it is a waste of resources as their intelligence should be reserved for high-value activities. Since the task is repetitive, human is also very likely to commit a mistake due to fatigue (Siddiqui et al., 2016). Moreover, the influx of a high volume of vehicle data can be overwhelming and it is not possible to analyze every piece of information in a fast, efficient, and accurate manner (Shokravi et al., 2020b; Siddiqui et al., 2016). Despite these, vehicle recognition also requires substantial knowledge to classify the vehicle correctly into its finest hierarchy. Hence, developing an Artificial Intelligence (AI)-based vehicle recognition solution is a compelling need.

AI-based solutions are normally built upon sophisticated sensing or Computer Vision (CV) technologies. Those off-the-shelf sensing devices require high installation costs and they are unscalable economy-wise. In addition, the stringent operating conditions must be met to produce accurate results. For instance, a loop detector is less reliable under high traffic volume (Sun & Ban, 2013) whereas an acoustic sensor often fails when the vehicles are overlapped. Some of these devices also require periodic maintenance.

Due to the shortcomings of sensing devices, the CV-based vehicle recognition system has become the primary choice (Dong et al., 2015; Y. Li et al., 2018; Wen et al., 2015) and the advancement is striking owing to the collective efforts of CV practitioners. As compared to sensing devices, CV-based techniques have loose operating conditions. The minimum ask is having access to the good-quality images for analytics purposes. License Plate Recognition (LPR) is one of the foremost applications that uses CV-based techniques to retrieve the identity of vehicles (Martín & Tosunoglu, 2000). For instance, Castello et al. (1999) performed LPR on the passing by vehicles. An alarm signal is sent to the control station if the license plate has been blacklisted. Without controversy, provided the LPR is accurate, all sorts of information such as brand, model, colour, manufacturing year, etc., for a vehicle can be retrieved easily from the authority's database. However, the prerequisite to the successful retriever of such information is the vehicle must be registered with the government of the respective country beforehand. Therefore, no information is available for foreign vehicles. In addition to that, the LPR is prone to error due to character ambiguity. For instance, '0' and 'O' are often appeared to be alike with each other (Siddiqui et al., 2016). The identification of vehicles through LPR is also bound to fail when the broken license plate has missing digits and alphabets. What is more discouraging is during the criminal investigation, the vehicle identity pinpointed through LPR may be less reliable as the license plate is normally forged (H. Liu et al., 2016; Martín & Tosunoglu, 2000; Siddiqui et al., 2016).

To march towards a forging-proof system, the proposed vehicle recognition solution should identify the vehicle based on its features. This is attainable through Vehicle Type Recognition (VTR), Vehicle Logo Recognition (VLR), Vehicle Make Recognition (VMR) and Vehicle Make and Model Recognition (VMMR). As illustrated in Figure 1.1, VTR allows the general categories of vehicles such as buses, lorries, Sports Utility Vehicle (SUV), sedans, etc., to be identified (Arinaldi et al., 2018; Dong et al., 2015; S. Li et al., 2018; Y. Li et al., 2018). VLR identifies the vehicle make based on the vehicle logo mounted on the hood and bonnet area (Cyganek & Woźniak, 2014; Huang et al., 2015). Based on the entire vehicle look, VMR and VMMR determine the vehicle identity at different granularity levels, which are automobile maker and vehicle model, respectively. Given the ability to recover more vehicle information, there is an increasing propensity towards VMMR due to its ability to obtain the finest information (Fang et al., 2016; Manzoor & Morgan, 2017; Satar & Dirik, 2018; Siddiqui et al., 2016).

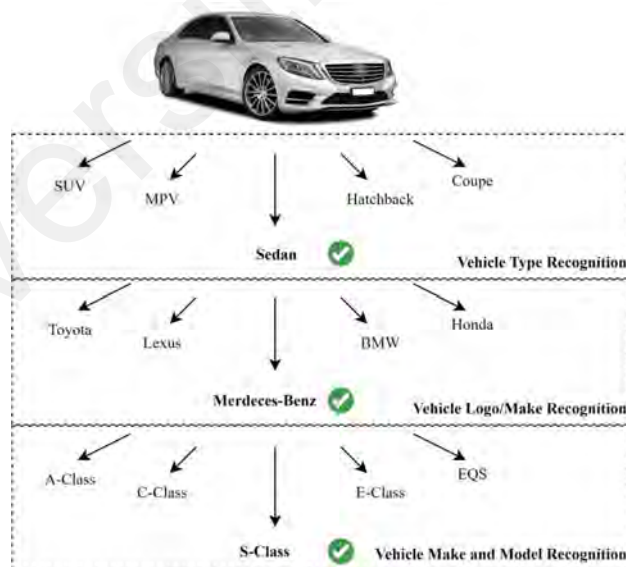


Figure 1.1: Illustration of Vehicle Type, Vehicle Logo, Vehicle Make and Vehicle Make and Model Recognition

Vehicle recognition is an intractable task to solve due to high intraclass variance and high interclass similarity. There are plenty of automobile makers out there and the

scenario becomes intricate when there are also many vehicle models under a make (Yang et al., 2015). Every vehicle model is unique in terms of shape, colour, size, etc (Wen et al., 2015). The complexity is termed multiplicity and ambiguity (Siddiqui et al., 2016). To explain further, multiplicity refers to the case where vehicles of the same model but different release years share similar designs. Ambiguity can be divided into two, namely inter-make and intra-make ambiguity. Inter-make ambiguity is portrayed as a phenomenon in which vehicles from different makes have minimal differences in their external looks whereas intra-make ambiguity refers to similar appearance among vehicles from the same make but different models. Due to all these challenges, the vehicle recognition solution must be designed meticulously so that it can learn an accurate decision boundary to achieve exceptional classification purpose.

1.2 Problem Statement

Vehicle recognition has undergone rapid development over the years. It started with handcrafted features such as Histogram of Oriented Gradient (HOG), Scale Invariant Feature Transform (SIFT) (Lowe, 2004) and Speeded Up Robust Feature (SURF) (Bay et al., 2008), evolved into feature encoding schemes and eventually being researched intensively in the Deep Learning (DL) domain. Over the years, there have been several well-renowned technical reviews in the vehicle recognition domain (Bernas et al., 2018; Boukerche et al., 2017; Datondji et al., 2016; Jain et al., 2019; Kul et al., 2017; Shokravi et al., 2020b; Tian et al., 2011; Won, 2020; Xie et al., 2021; Yousaf et al., 2012) but they are either less comprehensive or less relevant considering the published timeline. Therefore, there is a need to review, collate and consolidate the works, including the sensor-based and CV-based solutions, developed in recent years. These solutions deliver astounding performance and they possess high reference value. By comparing and contrasting these works, their strengths and drawbacks can be identified explicitly and this is beneficial to the proposal of a stronger and more competitive network.

Convolutional Neural Network (CNN) is a DL algorithm to solve problems in the spatial domain. Deducing convolution filters through backpropagation results in a long training duration. A popular way to overcome the weakness is to optimize them through unsupervised algorithms. For vehicle recognition, Sparse Laplacian Filter Learning (SLFL) (Dong et al., 2015), Principal Component Analysis (PCA) (Huang et al., 2015) and Topographic Independent Component Analysis (TICA) (Gao & Lee, 2016) were explored. Nevertheless, they did not resort to supervised techniques in learning the convolution filters and this renders the learned filters class-insensitive. The ground truth labels which detail the class separation information should be utilized to produce superior convolution filters.

Most of the CNNs are suboptimal for vehicle recognition tasks as they treat the neuron activations on the feature maps indistinctively. This is detrimental as every spatial position or channel information corresponds to different vehicle parts and the underlying information has different significance levels. Attention is an active ongoing research area in both Natural Language Processing (NLP) and CV in which the salient features are exploited to enhance the classification performance. Attention modules can be categorized into hard and soft attention. The shortcoming of hard attention is it drops seemingly unimportant information completely, and this hampers the classification performance if the decision is wrong. On the contrary, soft attention upweights and downweights the features according to their importance instead of discarding them fully. Although plenty of attention modules (Hu et al., 2018; Park et al., 2018; Wang et al., 2020) have been proposed for general classification tasks, the application of attention modules in the vehicle recognition domain is still limited. In addition, there are no strong fundamentals to date to guide the design of attention modules. It is believed that by incorporating an attention module that is customized for vehicle recognition tasks, significant improvement in classification accuracy can be observed.

Owing to the capability of DL to learn highly robust features, a significant portion of the research advances in the vehicle recognition domain hold on to CNNs. Despite their competitive performances, most of them follow the practice of general classification (Chollet, 2017; He et al., 2016; Huang et al., 2017; Krizhevsky et al., 2012; Sandler et al., 2018; Tan & Le, 2021) where only the top-level feature maps are used to infer classification logits. Such action compromises the feature representation ability of CNNs because an accurate classification is dependent on the availability of fine-grained subtle details of the object. It is advocated that the feature maps from various pyramidal levels should be used concurrently since they complement each other. The high-level global information, when backed by low-level local details, is believed to be able to enrich the feature embeddings and thus push the classification performance to a higher level. Several studies (Ding et al., 2021; Du et al., 2020; M. Liu et al., 2021) have shown exemplary performance by steering the research in this direction. Nevertheless, their ways of integrating the hierarchical features are not flawless as the global features overshadow the local features in the final representations and the cross-scale feature transfer method is simplistic. This leaves a question of how best these hierarchical features can be consolidated to achieve a balanced mix with minimal increase in computational cost.

1.3 Objectives

Based on the problem statements, this study aims to present an intelligent vehicle recognition system using the DL techniques. The research objectives are as follows:

- i. To deduce convolution filters through supervised and unsupervised dimension reduction techniques
- ii. To introduce a novel attention module that is equipped with the global field of view and is compatible with various CNNs for saliency-based feature refinement purpose

- iii. To propose modular multi-scale feature integration modules that bring positive impacts on classification performance through the exploitation of coarse-to-fine features

1.4 Scope of Research

Vehicle recognition is an interesting research domain that looks for viable ways to distinguish vehicles at various granularity levels. In this study, several vehicle recognition solutions that work for indoor and outdoor environments are devised. They are expected to process any given vehicle image and produce an accurate prediction in terms of vehicle type or vehicle model. To reach the end state, multiple extensive experiments that investigate supervised filter learning techniques, attention mechanisms and multi-scale features are conducted to design novel solutions that raise the feature expressive ability of the vehicle recognition framework. During the solution design stage, consideration is given to both the recognition performance as well as the inference speed for a balanced trade-off between the two. For performance benchmarking purposes, the study presents a thorough evaluation by comparing the proposed solutions against the existing state-of-the-art networks via a set of classification metrics. Additionally, the computational complexity is quantified by the number of parameters and floating point operations (FLOPs) as an indication of the ability to render real-time prediction. Apart from that, it is also the interest of the study to examine the generalization ability of the proposals to ensure they are highly compatible with various backbone networks.

An accurate and reliable vehicle recognition system forms the foundation for a handful of applications in ITS. Although the published research issues elucidate numerous invaluable and practical ideas to improve the performance of vehicle recognition, there is a lack of comparison and contrast among every framework. Thoroughly reviewing the existing works is beneficial since it helps to unveil new insights and useful techniques

across different spectrums of works and eventually inspires the conceptualization of novel techniques that further advance vehicle recognition performance.

To shorten the training process of CNNs, unsupervised filter learning techniques have emerged as an alternative to backpropagation. However, optimizing the convolution kernels without referring to label information may result in feature embeddings that are relatively indifferent to the class boundary. To rectify this, the supervised dimension reduction technique is exploited where it harnesses the label information to learn superior convolution filters. In the proposed shallow CNN architecture, the filters learned through unsupervised and supervised filter learning techniques complement each other to produce more discriminative feature representations that excel in vehicle recognition tasks.

Every feature response on the feature maps embeds information with varying degrees of importance. Disregarding the feature distinctiveness undermines the learning outcome of the network. Therefore, it is essential to allocate an adequate amount of attention that is proportionate to the information saliency so that the network bases its learning more on the crucial vehicle traits. An attention module is devised in this study to reweigh the activation values of the feature maps based on spatial relevancy. The eventual outcome is to empower the significant features in dominating the training process while keeping the influence of insignificant features at a minimal level.

The existing practice of ingesting top-level feature maps and sidelining shallow-level information in logit generation limits the representation capacity of CNNs. While the semantically strong deep-level information has a full grasp of the global information about the vehicle, the features of the vehicle can be enriched further by factoring in the shallow-level information since it focuses on the minute details that carry the differentiating factor for similar vehicle models. Therefore, it is the scope of this study to

investigate a mechanism to amalgamate multi-scale information to render more accurate predictions at the vehicle model level.

Several research assumptions have also been made while defining the scope of this study. Firstly, as performance benchmarking is conducted on public datasets, it is assumed that the provided annotations are correctly labeled. The datasets are expected to fulfill the minimum quality where the image resolution is at least 256×256 pixels so that the critical vehicle details are embedded within. Most important of all, the datasets are representative of real-life scenarios and have high diversity. There should be an adequate number of vehicle models captured under various viewpoints and environmental conditions to render a good simulation for the actual case and this ensures the relevancy of the study.

1.5 Significance of Research

Critical analysis of the existing works in the vehicle recognition domain helps to identify the advantages and disadvantages of various methodologies at a glance. Categorizing the published works into various streams according to the nature of the proposed algorithms allows the compare and contrast process to be carried out fairly and efficiently. The insightful information obtained from the review also accelerates the design of novel DL architecture by improvising the existing weaknesses.

The training duration of a CNN can be considerably long given the huge number of parameters. Optimizing the convolution kernels via unsupervised techniques in place of backpropagation has proven to be time-effective. Coupling the supervised dimension reduction techniques with the unsupervised counterpart during the filter learning process strengthens the vehicle recognition ability of the network even further and thus raises the classification performance.

Incorporating attention mechanisms into the feature map processing is essential in assuring the prioritization of prominent vehicle information. On one hand, it allows the training process to be properly guided by the crucial information. On the other hand, it ensures the inconsequential information has a minor contribution and yet provides the necessary contextual information regarding the object. The proposed attention module measures the feature relativity globally and scales the feature maps based on spatial importance. It eventually leads to a commendable learning process.

Multi-scale features offer a precious piece of information for vehicle recognition. The retention of such information is contributed by the multi-scale feature synthesis module that consolidates the feature maps produced by a series of convolutions. The resultant feature maps comprise local and global cues that are specific to a vehicle model. This study demonstrates that upon inserting the multi-scale feature learning module into the backbone CNN, the feature extraction ability is elevated and state-of-the-art results are reported. The proposed network is impactful not just in vehicle recognition, it can also be deployed to solve problems in other domains.

1.6 Thesis Outline

The thesis consists of a total of 8 chapters. They are the introduction, literature review, 4 research articles which encompass methodology, results and discussions and lastly conclusion.

Chapter 1 INTRODUCTION: It introduces the concept of ITS and how vehicle recognition underpins the success of such a vision. The research problems of vehicle recognition are presented and they are associated with the research objectives. The scope and significance of the research are explained in great length.

Chapter 2 LITERATURE REVIEW: This chapter unravels the relationship between AI, Machine Learning (ML) and DL. It also presents various types of DL architectures to solve problems from different domains. Besides, a comprehensive explanation of the components of CNN together with the well-renowned pretrained CNNs is deliberated. Most importantly, this chapter provides a thorough review of the existing vehicle recognition frameworks by dissecting the strengths and weaknesses of each work. A published version of the literature review is available in ‘Artificial Intelligent Systems for Vehicle Classification: A Survey’.

Chapter 3 PCA-LDA-BASED CONVOLUTIONAL NEURAL NETWORK WITH CHANNEL-BASED ATTENTION MODULE FOR VEHICLE MAKE AND MODEL RECOGNITION: In correspondence with the research objectives (i) and (ii), the supervised and unsupervised dimension reduction techniques i.e. PCA and Linear Discriminant Analysis (LDA) are employed as a substitution for the backpropagation technique in learning the convolution kernels. A parameter-free attention module, namely the Channel-Based Attention Module (ChBAM), is also inserted into the CNN for saliency-based channel refinement purposes.

Chapter 4 SPATIALLY RECALIBRATED CONVOLUTIONAL NEURAL NETWORK FOR VEHICLE TYPE RECOGNITION: Addressing the research objective (ii), an attention module dubbed Spatial Attention Module (SAM) powered by Multi-Head Self-Attention (MHSA) is studied. It refines the feature responses of the top-level feature maps to allow the significant vehicle parts to play a greater contribution in the training process. The design of SAM is justified through a series of in-depth ablation studies. This work is also described in the article ‘Spatially Recalibrated Convolutional Neural Network for Vehicle Type Recognition’.

Chapter 5 CROSS-GRANULARITY NETWORK FOR VEHICLE MAKE AND MODEL RECOGNITION: Conforming to the research objective (iii), this chapter investigates the generation of cross-granularity features for the enrichment of feature embeddings. The study culminates in the proposal of the Cross-Granularity (CG) module that presents a balanced mix of macroscopic and microscopic elements and is highly compatible with existing CNNs.

Chapter 6 COARSE-TO-FINE CONTEXT AGGREGATION NETWORK FOR VEHICLE MAKE AND MODEL RECOGNITION: Aligning with the research objective (iii), a lightweight multi-scale features module is expounded in this chapter. The Coarse-to-Fine Context Aggregation (CFCA) module employs multi-stream dilated convolution and recurrently performs information consolidation at various scale levels to render comprehensive multi-granularity feature maps for VMMR. A copy of the work is available in ‘Coarse-to-Fine Context Aggregation Network for Vehicle Make and Model Recognition’.

Chapter 7 AUGMENTED-GRANULARITY NETWORK FOR VEHICLE MAKE AND MODEL RECOGNITION: Contributing to the research objective (iii), this chapter unveils a highly competent multi-scale feature synthesis module called Augmented-Granularity (AG) module. The AG module employs grouped focus convolution (GFConv) that performs spatial-to-channel transformation to expand the field of view of convolution kernels for the learning of holistic vehicle features. An extension analysis demonstrates the efficacy of the AG module.

Chapter 8 CONCLUSION: This chapter summarizes all the research works undertaken during the entire course of the study and provides suggestions for future research direction.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

Computer Vision (CV) is a field of Artificial Intelligence (AI) that endows a machine with a sense of sight so that it acquires the capability of conceiving objects like human beings. The sense of sight is the outcome of a series of steps including image acquisition, image labeling, image processing, formulation of network architecture, training, validation, hyperparameter tuning, etc. All these steps ultimately determine the quality of sight that a machine can gain.

Generally, a CV task can be segregated into a classification, detection and segmentation problem. Classification refers to the categorization of an object in the image to its class. Detection is more complex since aside from determining the objects in the image, it provides the location of the objects in terms of pixel coordinates. Segmentation is similar to classification but it performs classification at pixel level to achieve complete isolation of objects between one another. Aside from the above-mentioned, CV problems cover reidentification, noise removal, image generation, etc.

The value brought by CV is unbounded in which it is used by most industries to elevate working efficiency and achieve cost optimization. The areas of applications include Intelligent Transportation System (ITS) (Alghamdi et al., 2023; Basheer Ahmed et al., 2023; Cao et al., 2023; Chen, 2023; Qibtiah et al., 2023), defects detection (Farady et al., 2023; Guan et al., 2023; Lin et al., 2023; Nath et al., 2023; Perri et al., 2023), medicine (Bortoluzzi et al., 2023; Hao et al., 2023; Ismael & Şengür, 2021; Karthik & Mahadevappa, 2023; G. Li et al., 2023), agriculture (S. Chen et al., 2022; Grijalva et al., 2023; Gulzar, 2023; Pal & Kumar, 2023; Paymode & Malode, 2022), security (Hu et al., 2020; Jyothi et al., 2023; Kumar & Janet, 2022), etc.

2.2 Machine Learning

Machine Learning (ML) is one of the means to implement CV and it is a subset of AI and computer science as depicted in Figure 2.1. It imitates the human learning process by uncovering patterns in the data with the help of algorithms to better associate the input and output signals. As early as 1959, ML came to light. Samuel (1959) conducted a study on checker games using it. In 1962, the defeat of Robert Nealey, the checker master, was seen during his competition with the IBM machine. Ever since then, ML is highly sought after to solve real-world problems.

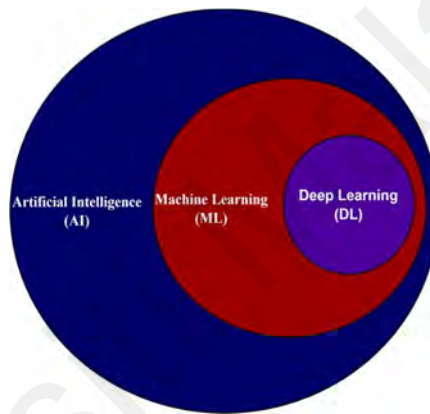


Figure 2.1: Artificial Intelligence, Machine Learning and Deep Learning

ML algorithms encompass three components. They are the decision process, error function and optimization process. These three components form a closed-loop system where correction is done based on the feedback given to the machine's response. In simpler words, a decision process takes in the data and produces a prediction. The goodness of prediction is then quantified by comparing the actual and predicted responses through the error function. Lastly, an optimization process rectifies the mistake and the decision logic is updated to achieve more accurate prediction in the subsequent iteration. The cycle of the decision process, error function and optimization process runs indefinitely until the stopping criteria are met in which the error is negligibly small or the maximum number of iterations is reached.

There are various types of ML algorithms that one can choose depending on data availability. Supervised learning ingests the data with ground truth labels and the performance of the algorithm is evaluated based on a set of unseen data. Unsupervised learning is selected when the data is unlabeled and the algorithm is expected to uncover patterns and to learn decision logic without the help of labels. Semi-supervised learning is a hybrid of both where it augments the size of training data by first training the model on data with known labels and subsequently uses the first-cut model to complete the labeling process. The last type of learning is reinforcement learning which involves the concept of reward and punishment to allow the algorithm to learn from experience through trial and error.

Under every learning regime, there are multiple algorithms at one's disposal. Each algorithm produces different learning results given the same dataset since the underlying assumptions of the algorithms are different. Hence, it takes experience and a thorough understanding of the data to land on the right algorithm. The common ML algorithms are linear regression, logistic regression, decision tree, random forest, gradient boosting, support vector machine (SVM) and k-Means.

2.3 Deep Learning

Although ML and Deep Learning (DL) are often used interchangeably, they are different in subtle ways. DL is a subset of ML and it is known for its automated feature engineering. During the training process, a DL algorithm i.e. neural network derives new features and performs feature weighting based on the backpropagated loss. On the contrary, a ML algorithm requires the discretion of human experts to determine the set of features that works best in learning the mapping between input data and label.

The history of DL can be dated back to 1943 when McCulloch and Pitts (1943) proposed the first-ever artificial neuron called the MCP neuron. The artificial neuron is

an inspiration by biological neurons in the human brain that serve as the building block to resolve high-complexity problems. As portrayed in Figure 2.2, a biological neuron receives the electrical signals from the neighboring neurons through dendrites. The signals are then aggregated within the soma and it is passed on to the axon hillock. If the magnitude of the signal is strong, the neuron is activated. The neuron state is also propagated to other neurons through synapses.

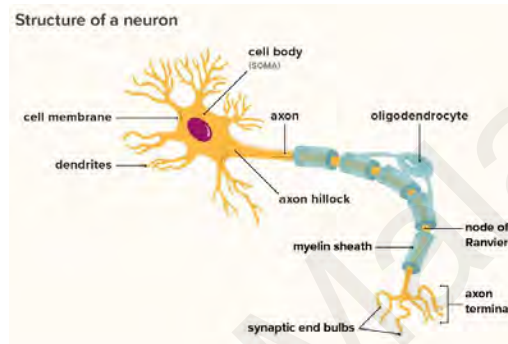


Figure 2.2: Biological Neuron (Vandergrindt & Zimlich, 2022)

Later in 1958, Rosenblatt (1958) made an improvisation to the MCP neuron by proposing perceptron. Perceptron relaxes the absolute inhibition, weighs the input signals based on significance and possesses the ability to accept floating point value. The revised architecture serves as the fundamental component of the contemporary neural network. As shown in Figure 2.3, a perceptron performs a weighted combination of the input signals X_i and an activation function is applied subsequently to model the non-linear relationship between input and output signals.

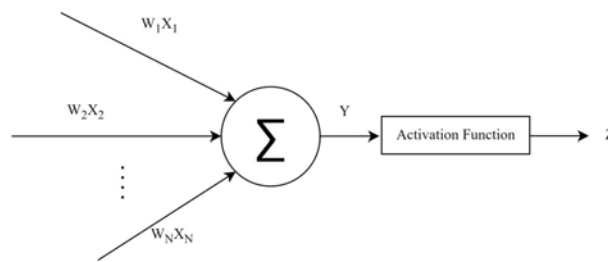


Figure 2.3: Perceptron

The operation within a perceptron is mathematically represented as

$$Y = \sum_{i=1}^N X_i W_i + b \quad (2.1)$$

where W_i is the weight applied on X_i , b is the bias, N is the number of input signals and Y is the resultant linear response.

To inject non-linearity into the signal, an activation function follows. The common activation functions are sigmoid, hyperbolic tangent and rectified linear unit (ReLU) (Nair & Hinton, 2010) as formulated in Equation 2.2 – Equation 2.4, respectively.

$$Z = \frac{1}{1 + e^{-Y}} \quad (2.2)$$

$$Z = \tanh(Y) = \frac{e^Y - e^{-Y}}{e^Y + e^{-Y}} \quad (2.3)$$

$$Z = \max(0, Y) \quad (2.4)$$

As sigmoid and hyperbolic tangent limit the output signal between $[0, 1]$ and $[-1, 1]$, respectively, they pose vanishing gradient problems and this inhibits the neurons from learning. To overcome the shortcoming, ReLU is widely adopted where it acts as a high pass filter that suppresses the negative signal and the gradient is kept in large value. In addition, the complexity of ReLU is much lower due to the elimination of the exponential function. This is reflected in faster forward passes and shorter training duration. Since the performance brought by ReLU is remarkable, more variants are proposed and they are Leaky ReLU (Maas et al., 2013), Parametric ReLU (K. He et al., 2015), Gaussian Error Linear Unit (GeLU) (Hendrycks & Gimpel, 2016), etc.

The representation learning of a single perceptron unit can be elevated further by aggregating several of them to form a Multi-Layer Perceptron (MLP). Figure 2.4 depicts the structure of MLP which consists of input, hidden and output layers. The number of perceptrons within the hidden layer is an important design parameter since setting an optimum number leads to a superior learning process. When the number of hidden layers is larger than one, a more complex version of MLP is formed and such structure is dubbed deep neural network.

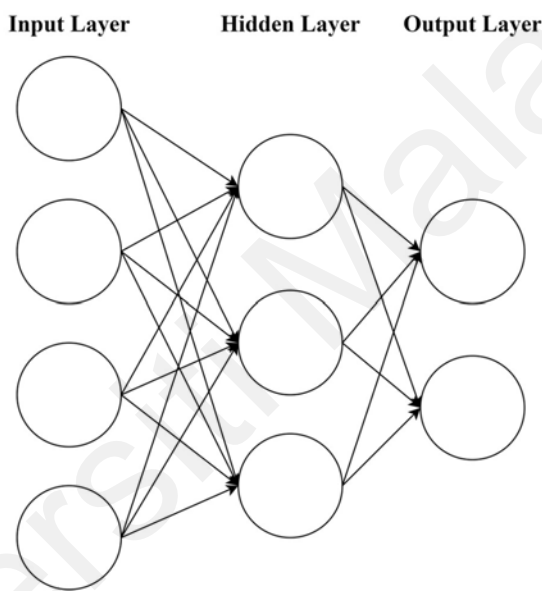


Figure 2.4: Multi-Layer Perceptron

Although the neural network was first proposed half a century ago, it only attracted lots of attention from both researchers and practitioners in recent decades. In the past, it was beheld by the scarcity of data and expensive hardware. Given the availability of large-scale datasets such as ImageNet (Deng et al., 2009), iNaturalist (Horn et al., 2018) and JFT-3B (Zhai et al., 2022) nowadays as well as the reduction of hardware costs especially Graphic Processing Unit (GPU), the full potential of DL is unlocked and its impressive performance in regression, classification, Natural Language Processing (NLP) and CV domain is seen.

2.3.1 Recurrent Neural Network

In the structured data domain, the deep neural network is the primary choice of many due to its compelling performance. However, it does not work well with temporal data such as text, images and audio. It treats every piece of information independently, hence it fails to capture the sequence relationship. A Recurrent Neural Network (RNN) is an effective architecture for learning sequential data. By utilizing the memory cell to retain the previous inputs, a more meaningful representation is derived. The dependency on prior elements is important as it allows the network to encode the information within the right context. The bulk of RNN is composed of recurrent units called Long Short-Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997) and Gated Recurrent Unit (GRU) (Cho et al., 2014). The architectures of LSTM and GRU are portrayed in Figure 2.5. They are effective in associating important states together even at distant locations to a certain extent. LSTM contains three types of gates, namely input, output and forget gate. These gates collaborate to perform information filtering by storing prominent information in the cell states. As compared to LSTM, GRU is more lightweight since it uses fewer gates i.e. reset and update gates to regulate the information.

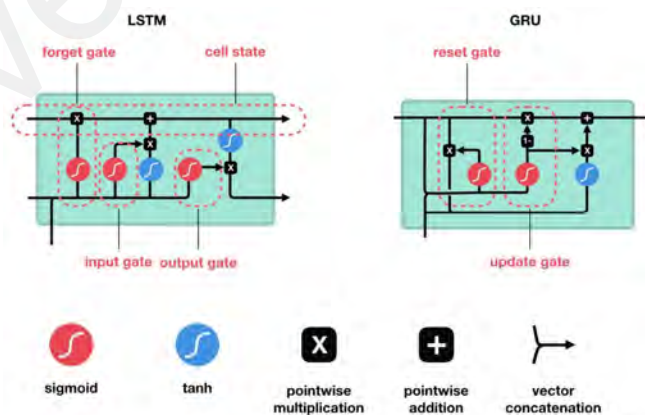


Figure 2.5: Long Short-Term Memory and Gated Recurrent Unit (Phi, 2018)

2.3.2 Convolutional Neural Network

Convolutional Neural Network (CNN) is the dominant architecture in solving tasks related to CV and audio. Referring to Figure 2.6, it has five types of layers, namely the convolution, normalization, activation, pooling and fully connected layer.

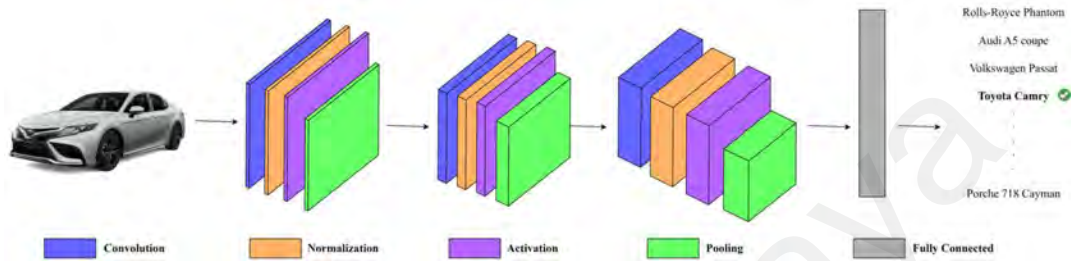


Figure 2.6: Convolutional Neural Network

Essentially, a convolution operation applies convolution kernels on the input pixels to produce output features and the kernels are slid until the entire image is covered. The resultant matrix produced from the convolution operation is called a feature map. Convolution works based on the principles of local connectivity and parameter sharing (LeCun et al., 1989). Local connectivity symbolizes that convolution kernels convolve within the local region whereas parameter sharing signifies that the same convolution kernels are applied across the entire feature maps. Such principles reduce the number of trainable parameters in CNN drastically while promoting translation invariance. It is worth noting that the convolution operation learns features from the images in stages. The early convolution focuses on the raw details of an object such as edges, corners and texture information and these are known as low-level features. As the convolution operation progresses, high-level features with deep semantic understanding are deduced and they are useful for object recognition. The convolution operation is mathematically represented as follows:

$$I_{conv} = Conv_{k \times k, C_{out}, s}(I_i) \quad (2.5)$$

where I_{conv} is the feature maps post-convolution, $Conv_{k \times k, C_{out}, s}(\bullet)$ is s -strided $k \times k$ convolution with C_{out} output filters and I_i is the input image.

Modern CNNs often insert batch normalization (BN) (Ioffe & Szegedy, 2015) as the normalization layer before the activation layer to address the issue of internal covariate shift. Covariate shift is the significant shift in distribution among different batches of data in neural networks. It hinders the learning process, which is reflected explicitly in terms of model training time as the model needs to juggle between distributions that are different from one batch to another. Ioffe and Szegedy (2015) also demonstrated that BN can play a regularization role in model training and it leads to accuracy improvement.

The role of an activation layer is to induce non-linearity to help the network model the non-linear relationship between input and output. Thereafter, the pooling layer downsizes the feature maps. The pooling layer applies average or maximum aggregation in a sliding window manner and it involves no training parameter. Average pooling computes the mean response whereas maximum pooling preserves the most salient information within the receptive field. In modern CNN architecture, convolution-based pooling is widely adopted where it downsizes the feature maps through a strided convolution but this comes with minimal computational cost. Although pooling results in information loss, it leads to higher training efficiency by encapsulating the feature responses in smaller feature maps. Pooling operation is mathematically written as

$$I_{pool} = Pool_{k \times k, s}(I_i) \quad (2.6)$$

where I_{pool} and $Pool_{k \times k, s}(\bullet)$ are the resultant feature maps and pooling operation, respectively.

2.3.3 Transformer

In recent years, the development of RNN and CNN architectures has converged following the advent of transformers. Transformer is a universal architecture that suits text, image and audio domains. Regardless of the domains, it processes the inputs as a sequence of tokens with a few encoder layers that consist of Multi-Head Self-Attention (MHSA) and Feed Forward Network (FFN). The MHSA first computes the compatibility between query and key matrices in several feature subspaces before altering the value matrix based on the significance of underlying information. The FFN is a two-layer neural network that contributes to capturing the intricate and non-linear relationship of the inputs. At the end of the network, the class token, which is prepended to the input sequence, is utilized to infer logits.

2.3.4 Fully Connected Layer

For the classification task, the RNN, CNN and transformer architectures are ended with a fully connected layer. A fully connected layer is a deep neural network that is responsible for computing class-wise probability upon applying the softmax function. Since no parameter-sharing strategy is adopted here, the number of trainable parameters of the network mostly comes from this layer. Given L number of classes, the probability of an object belonging to class l can be computed from l^{th} output neuron x_l from the fully connected layer using the softmax function.

$$P(\text{Class} = l | x_l) = \text{Softmax}(x_l) = \frac{e^{x_l}}{\sum_{i=1}^L e^{x_i}} \quad (2.7)$$

2.3.5 Loss Function

The loss function is used to quantify the goodness of fit of the trained network. It computes the degree of matching between prediction and ground truth labels. Partial differentiation is then computed with respect to each trainable parameter and the

parameters are updated to achieve convergence of the loss function. There are a lot of loss functions out there and an optimal loss function is much dependent on the dataset, network architecture and training methodology. Opting for the right loss function ensures superior network performance, faster convergence and optimal bias-variance trade-off.

Cross entropy loss is the common loss function for classification. It is given by

$$Loss = -\frac{1}{N_{Train}} \sum_{i=1}^{N_{Train}} \sum_{l=1}^L y_{i,l} \log p_{i,l} \quad (2.8)$$

where N_{Train} is the number of training samples, $y_{i,l}$ is a binary indication of image i belonging to class l and $p_{i,l}$ is the class probability.

One thing to note about cross entropy loss is it trains the network to pursue perfect prediction and this may lead to overfitting. In addition, it is built based on the assumption that all the data is labeled correctly which is hardly the case for large-scale datasets. As a result, a label smoothing factor (Szegedy et al., 2016) is introduced where a uniformly distributed noise is injected to prevent the network from being overconfident with the prediction.

$$Loss = -\frac{1}{N_{Train}} \sum_{i=1}^{N_{Train}} \sum_{l=1}^L (1 - \epsilon) y_{i,l} \log p_{i,l} + \frac{\epsilon}{(L - 1)} y_{i,l} \log p_{i,l} \quad (2.9)$$

where ϵ is the label smoothing coefficient.

Contrastive loss (Chopra et al., 2005) is a form of metric learning that optimizes the embeddings based on the target class. The goal is to create a feature space where the objects from the same class are in close vicinity as compared to that of different classes to make contrasting between similar and dissimilar objects easier. Contrastive loss is formulated by

$$Loss = \frac{1}{N_{Train}^2} \sum_{i=1}^{N_{Train}} \left[\sum_{j:y_i=y_j}^{N_{Train}} (1 - Sim(z_{emb,i}, z_{emb,j})) + \sum_{j:y_i \neq y_j}^{N_{Train}} \max(Sim(z_{emb,i}, z_{emb,j}) - \omega, 0) \right] \quad (2.10)$$

where y_i and $z_{emb,i}$ are ground truth and deep feature embeddings, $Sim(\bullet)$ is cosine similarity function and ω is a constant margin that sets the loss of the negative pair to 0 if the similarity score is low enough.

Another loss function called focal loss (Lin et al., 2017b) improves the learning process by emphasizing hard examples. By adding a modulation factor, the network is penalized more for producing a low classification score. On the contrary, the loss contributed by an easy example is minimal and this provides incentive to the network to correct the misclassification. The focal loss is given as

$$Loss = -\frac{1}{N_{Train}} \sum_{i=1}^{N_{Train}} \sum_{l=1}^L y_{i,l} (1 - p_{i,l})^\gamma \log p_{i,l} \quad (2.11)$$

where γ is the focal factor that decides the degree of penalization towards hard examples. When $\gamma = 0$, focal loss is equivalent to cross entropy loss.

The strategy of paying attention to hard examples also inspires the proposal of gradient-boosting cross entropy loss (Sun et al., 2020). However, the loss function is designed in such a way that top-k confusing classes are the only contributors. This design choice allows the weights of trainable parameters to be updated swiftly towards convergence as compared to the primitive cross entropy due to larger gradient information after partial derivative. Gradient-boosting cross entropy loss is represented as

$$Loss = -\frac{1}{N_{Train}} \sum_{i=1}^{N_{Train}} \sum_{l=1}^L y_{i,l} \log \frac{e^{p_{i,l}}}{e^{p_{i,l}} + \sum e^{p_{i,k}}} \quad (2.12)$$

where $\sum e^{p_{i,k}}$ is the summation of probability from top-k confusing classes.

2.3.6 Backpropagation

Backpropagation is a technique employed to propagate the loss with respect to the parameters. It produces the gradient and it is used to update the parameters to achieve minimal loss. Based on the commonly adopted loss function which is cross entropy loss, the partial derivative with respect to the l^{th} output neuron in the fully connected layer is generalized as

$$\frac{\partial Loss}{\partial x_l} = \sum_{i=1}^L y_i \cdot Softmax(x_l) - \sum_{i=1}^L y_i \cdot 1\{i = l\} \quad (2.13)$$

where $1\{i = l\}$ is an identity function that gives 1 when i is equal to l otherwise 0.

2.3.7 Optimization

Upon getting the gradient from backpropagation, the subsequent step is to update the network parameters with the help of an optimizer. Stochastic Gradient Descent (SGD) is a popular optimizer used in the training of neural networks and it originates from gradient descent algorithm. To allow the search for a global minimum with less disturbance from the oscillation of noisy gradients, SGD is coupled with momentum. The process to update trainable parameters θ is represented as

$$V_t = \beta V_{t-1} + (1 - \beta) \frac{\partial Loss}{\partial \theta} \quad (2.14)$$

$$\theta = \theta - \alpha V_t \quad (2.15)$$

where V_t is the gradient at iteration t upon applying exponential decay β and α is the learning rate.

Adaptive Moment Estimation (Adam) (Kingma & Ba, 2014) is an extension of SGD where it adaptively adjusts the learning rate to speed up the convergence of the loss function. Adam updates θ following the steps below.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \frac{\partial Loss}{\partial \theta} \quad (2.16)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) \frac{\partial Loss^2}{\partial \theta} \quad (2.17)$$

$$\hat{m}_t = \frac{m_t}{(1 - \beta_1^t)} \quad (2.18)$$

$$\hat{v}_t = \frac{v_t}{(1 - \beta_2^t)} \quad (2.19)$$

$$\theta_t = \theta_{t-1} - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \quad (2.20)$$

where β_1 and β_2 , are exponential decay rates for first and second moment estimates, β_1^t and β_2^t are exponential decay rate for first and second moment estimates raised to the power t , m_t , v_t , \hat{m}_t and \hat{v}_t are first and second moment estimates before and after bias correction and ϵ is a constant to avoid zero division error.

2.3.8 Pretrained Convolutional Neural Network

Pretrained CNNs refer to the networks that have been trained on large data regimes for general classification tasks. The pretrained weights provide a good initialization and they can be adjusted further during transfer learning or fine-tuning to suit the desired downstream tasks. Transfer learning and fine-tuning allow the network to achieve

remarkable performance swiftly within a shorter training duration as compared to that of random weight initialization. The pretrained CNNs also serve as the foundation for building a stronger network. For instance, Squeeze-Excitation Network (Hu et al., 2018), Feature Pyramid Network (Lin et al., 2017a) and DeepLabv3+ (Chen et al., 2018) are among the networks that use Residual Network (ResNet) (He et al., 2016) as the backbone. Building the network on a strong backbone CNN is the stepping stone to achieving state-of-the-art results. Hence, it is important to get to know different types of CNN architectures that have been published.

2.3.8.1 AlexNet

Krizhevsky et al. (2012) topped the leaderboard of the ImageNet Large-Scale Visual Recognition Challenge in 2012 with an 8-layer deep CNN called AlexNet. The architecture of AlexNet is shown in Figure 2.7.

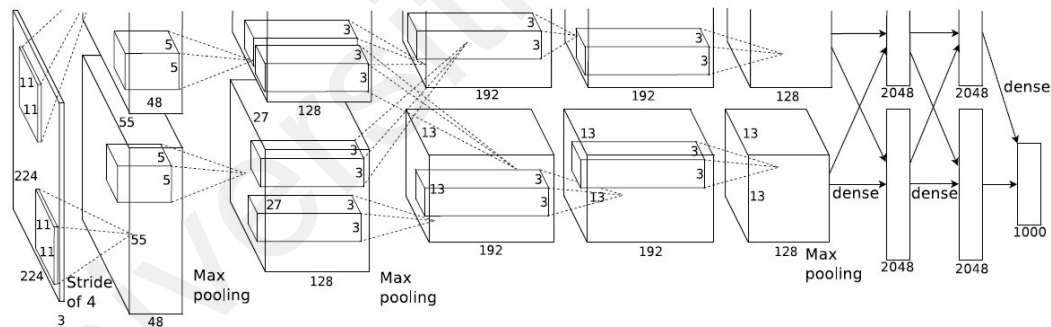


Figure 2.7: AlexNet (Krizhevsky et al., 2012)

It consists of 5 convolution layers and a 3-layer fully connected layer. ReLU is used as the activation function to speed up the training and dropout is used to randomly nullify the input neurons as a regularization technique. Apart from that, it features local response normalization (LRN) to perform normalization based on the neighboring neurons.

2.3.8.2 VGG

VGG (Simonyan & Zisserman, 2014) is a lightweight neural network as compared to AlexNet. Instead of using the parameter-heavy large convolution kernel, it uses 3×3 convolution kernels to produce the pyramidal features. This provides VGG the luxury of having 16 layers (138M) and even 19 layers (144M) as compared to 8-layer AlexNet (62.3M). Having more convolution layers allows VGG to produce more distinctive features, hence rendering better classification performance.

2.3.8.3 Inception Network

Inception Network utilizes the Inception module to produce pyramidal features. An Inception module contains various sizes of convolution kernels to perform feature extraction from objects of various sizes. To save the computational cost, large convolution kernels are factorized into multiple 3×3 convolutions and asymmetric convolution is also introduced where $k \times k$ convolution is broken down into $1 \times k$ and $k \times 1$ convolution. Since its inception, it has been revised continuously into four different versions. The latest version of Inception Network is Inceptionv4 (Szegedy et al., 2017) which features more Inception modules than its predecessor to enrich the feature embeddings. The architecture of Inceptionv4 is illustrated in Figure 2.8.

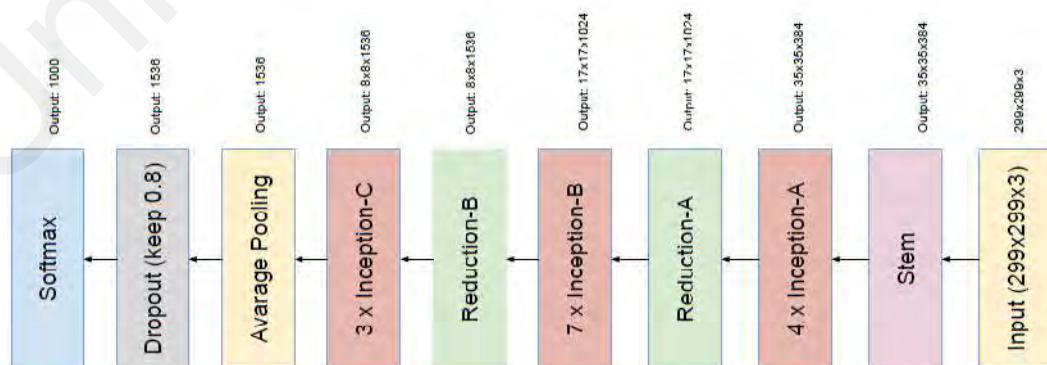


Figure 2.8: Inceptionv4 (Szegedy et al., 2017)

2.3.8.4 Residual Network

The development of VGG from AlexNet has proven that adding more convolution layers increases the classification performance. Nevertheless, this is only valid to a certain extent due to the vanishing gradient problem. To resolve this, He et al. (2016) inserted shortcut connections into the ResNet and this strategy ensures the deep CNN has an accuracy not lesser than its shallower counterpart. A few notable variants of ResNet are ResNet18, ResNet34, ResNet50 and ResNet101.

2.3.8.5 Densely Connected Convolutional Network

Densely Connected Convolutional Network (DenseNet) (Huang et al., 2017) employs dense connections where each layer receives the inputs from all preceding layers. The dense connection is effective for vanishing gradient problems just like ResNet's shortcut connection since it allows the gradients to be propagated directly to early layers. Such design also elevates the computational efficiency due to the lesser number of channels. More importantly, it ensures diversified learning where the shallow-to-deep features are used to learn the decision boundary.

2.3.8.6 Efficient Network

The proposal of the Efficient Network (EfficientNet) (Tan & Le, 2019) is motivated by the need to scale the network wisely in terms of depth, width and resolution without hitting the computational resource constraint. It is later revised into EfficientNetv2 (Tan & Le, 2021) to achieve higher training speed and parameter efficiency as well as accuracy. The amelioration includes adopting a hybrid of MBConv and Fused-MBConv as the building blocks to realize high-speed training. In addition, a progressive learning strategy is introduced where the regularization strength is adjusted according to the network and image size to achieve better learning experience.

2.4 Vehicle Recognition System

Generally, vehicle recognition can be addressed via either sensor- or CV-based approaches as shown in Figure 2.9. A sensor is defined as a device or module that measures the physical property of a substance by responding to stimuli such as sound, magnetism, light, etc. Within the sensor-based approach, the works can be further categorized based on the installation points, which are on-roadway, side-roadway and over-roadway (Won, 2020).

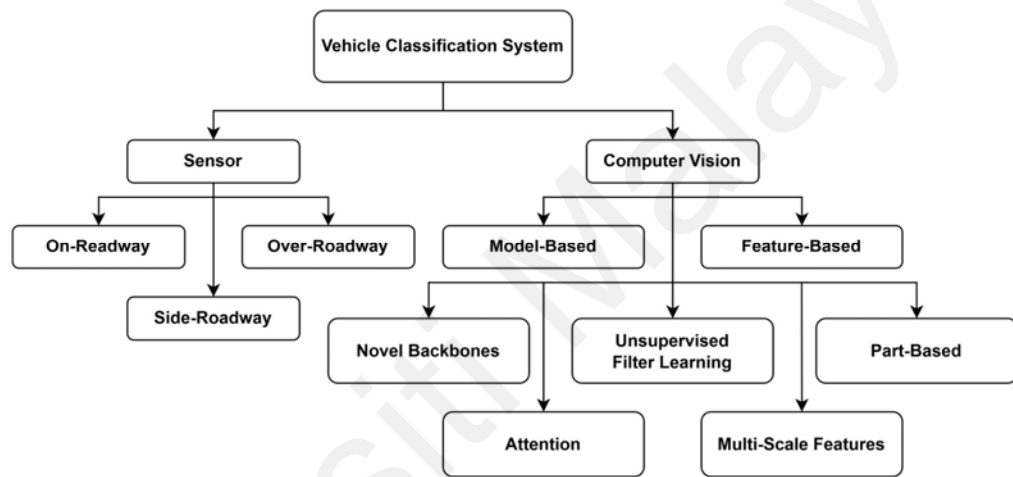


Figure 2.9: Taxonomy of Vehicle Recognition System

The CV-based approach refers to the solutions that receive RGB images as input. Based on the nature of the methodologies, they can be segregated into a total of 7 groups, covering model-based, feature-based, novel backbones, unsupervised filter learning, part-based, attention and multi-scale features.

Model-based method characterizes the vehicle shape using a 3D model based on 2D vehicle images collected from different viewpoints. Leveraging vehicle shape information has shown exemplary performance on various datasets.

Paying attention to the edges and corners information of vehicles also shows encouraging results. This is evident in the feature-based method which mines the changes

in pixel intensity to identify the gradient orientation and interest points. The extracted information is deemed distinctive as it helps ML classifiers perform the vehicle recognition tasks well.

Motivated by the astounding performance of AlexNet (Krizhevsky et al., 2012) in ImageNet competition, the research area is shifted to DL where CNN was the central of the studies during the early times. The core operating principle of CNN is to leverage convolution kernels in learning strong semantic information. Over the years, CNN architectures have undergone rapid evolution to raise the feature extraction ability while ensuring modest computational complexity. Due to the success of transformer architectures in the NLP domain, the transformer architectures are also gaining traction in the CV domain. They are known for the ability to track long-range dependencies via MHSA. Therefore, two development streams stemming from CNN and transformer architectures can be seen in the novel backbones category.

Since the computational resources required by CNN are not trivial, some studies look into unsupervised filter learning. It is a more economical way to learn the convolution filters without the expensive backpropagation technique. What is more motivating is the learned filters are more distinctive than those learned via backpropagation.

Without controversy, the image background carries noisy information that may divert the networks during the feature extraction stage. To improve feature representation learning, the part-based method focuses solely on the vehicle object by eradicating inconsequential information through localization. The localization can be learned by feeding the network with bounding box annotation or in a weakly supervised manner.

Instead of suppressing the background information fully, the attention method takes a slightly less extreme approach. It downweighs the irrelevant information with soft

attention masks to avoid the complete removal of contextual information. In other words, the attention masks perform feature recalibration so that the significant feature responses exercise more influence than the rest.

Multi-scale features method refers to a group of works that seeks ways to harness the hierarchical features produced from the image pyramids. The hierarchical features encompass both local fine-grained details and global abstract information. Learning the vehicle recognition task using the multi-granularity features results in comprehensive representation and it brings substantial improvement to the baseline network.

2.4.1 Sensor

2.4.1.1 On-Roadway

On-roadway sensors enjoy higher contact with the vehicles since they are usually buried under the road pavement. The installation work is relatively massive and it translates to higher expenditure. In addition, their reliability in terms of accuracy is reduced significantly under high traffic volume (Sun & Ban, 2013). Meta and Cinsdikici (2010) generated the magnetic profile of the vehicles via a single-loop detector. After denoising the raw magnetic signal with Discrete Fourier Transform (DFT), the principal components are extracted via Principal Component Analysis (PCA) and they are used to train the neural network. The single-loop detector is also utilized by Tok and Ritchie (2010) and the magnetic signal is further supplemented with the vehicle axle information. Jeng et al. (2013) applied the Haar wavelet transform and k-nearest neighbour (kNN) on inductive vehicle signatures drawn from the single-loop detector to differentiate 13 vehicle classes. The dual-loop detector leverages an additional loop to calculate the vehicle speed and length by studying the temporal difference of the signals. The vehicle length information deduced from the dual-loop detector is used to infer vehicle classes

(Li, 2010; Wei et al., 2013; Wu & Coifman, 2014). Although the dual-loop detector can render more vehicle information, it is more costly than the single-loop detector.

Another choice of on-roadway sensor is the magnetic sensor. As depicted in Figure 2.10, the magnetic sensor captures the change in the magnetic field induced by the metallic vehicle body to differentiate various vehicle types. Compared to the loop detector, it is more favorable, lightweight and economical. Furthermore, it is more robust against the Doppler effect and environmental disturbances. Dong et al. (2018) ingested the magnetic signal into XGBoost (Chen & Guestrin, 2016) to classify 4 vehicle types. Based on the vehicle length information derived from the magnetic signal, Balid et al. (2017) and Bottero et al. (2013) also delivered remarkable performances for Vehicle Type Recognition (VTR). The downside of the magnetic sensor is it requires intense calibration.

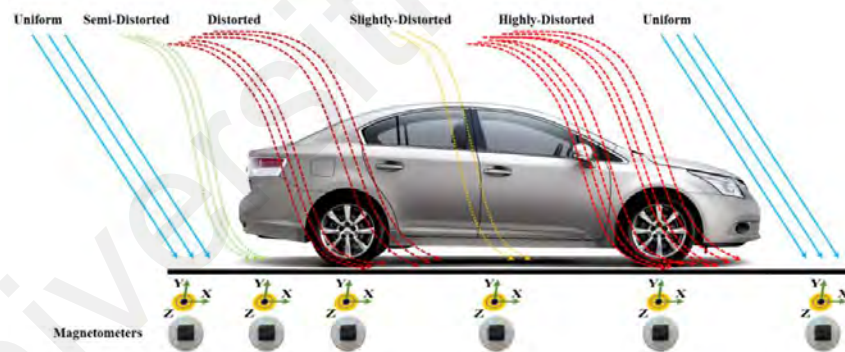


Figure 2.10: Operating Principle of Magnetic Sensor (Balid et al., 2017)

Rajab et al. (2016) derived several features, namely the number of tires, vehicle length, and axle spacing, from the vehicle signal measured by the piezoelectric sensor. As for Ahn et al. (2011), the sensor is used to infer vehicle weight. The performance of the piezoelectric sensor is very much dependent on the vehicle speed and road surface temperature.

A pneumatic tube sensor characterizes the vehicles through multiple rubber hoses that extend across the road. Its applications include vehicle counting (Nordback et al., 2016), traveling time estimation (Murrugarra et al., 2010). and vehicle speed and axle prediction (Gholamhosseinian & Seitz, 2021).

To encapsulate, the on-roadway sensors include detectors, magnetic sensors, piezoelectric sensors and pneumatic tubes. Owing to the physical contact with the vehicles, the deduced vehicle profile is more accurate but such an advantage may be potentially outweighed by the installation costs and inability to cater to high traffic volume. Table 2.1 provides a summary of the works powered by on-roadway sensors where accuracy is chosen as the default metric unless stated otherwise.

Table 2.1: Summary of On-Roadway Methods

Reference	Sensor	Method	Dataset	#Classes	Metric	Highlight
Meta and Cinsdikici (2010)	Single-loop Detector	DFT, PCA & Neural Network	Private	5	94.2%	DFT improves classification performance eliminating noisy signal in loop detector.
Tok and Ritchie (2010)	Single-loop Detector	Neural Network	Private	5	99.0%	The targeted classes are non-commercial vehicle types. The performance needs further examination as the experiment considered only a single lane.
				9	84.9%	
				10	84.1%	
Jeng et al. (2013)	Single-loop Detector	Haar Wavelet Transform & kNN	Private	13	93.8%	Determination of the signature template for each vehicle class is challenging but it is crucial to the classification performance.
Wu and Coifman (2014)	Dual-loop Detector	Length-based Classifier	Private	3	99.0%	Taking vehicle acceleration into consideration results in a more accurate estimation of vehicle length.
Dong et al. (2018)	Magnetic Sensor	XGBoost	Private	4	80.5%	Short-term temporal features are among the important features used in classification.
Bottero et al. (2013)	Magnetic Sensor	Length-based Classifier	Private	3	88.0%	Since every vehicle cannot travel at the same speed, the speed-dependent correction threshold and length threshold are hard to calibrate and it brings classification accuracy lower.
Balid et al. (2017)	Magnetic Sensor	C-SVM	Private	4	97.8%	Since vehicle magnetic length is not discriminative among different vehicle classes, differentiating 4 number of classes is most appropriate.
Rajab et al. (2016)	Piezoelectric	Naïve Bayes	Private	13	86.9%	The piezoelectric sensor must be placed at an optimum angle for best classification performance.
Nordback et al. (2016)	Pneumatic Tubes	Air Pulse-based Classifier	Private	2	84.9%	Accuracy decreases with increasing traffic volume.

2.4.1.2 Side-Roadway

Generally, the installation cost of side-roadway sensors is lower than on-roadway sensors since no saw-cutting of the road is required to bury the sensors. Nevertheless, they require precise calibration and the performances degrade severely under high traffic volume due to the blocked vehicle view (Sun & Ban, 2013). Bischof et al. (2010) implemented a semi-supervised learning based on acoustic sensors. The acoustic signals are used to annotate the vehicle type which is later used to train an ensemble of audio and visual-based classifiers. Acoustic sensors are also used for feature extraction purposes by Wiczorkowska et al. (2018).

Similar to acoustic sensors, Light Detection and Ranging (LiDAR) works based on signal reflections but it leverages laser beams instead. Based on the setup in Figure 2.11, Asborno et al. (2019) deployed a multi-LiDAR system to describe the vehicle. The features derived from LiDAR are used to train the Naive Bayes Ensemble. Radio Detection and Ranging (RADAR) is another alternative under side-roadway sensor. It operates on radio waves. Although the characterization is poorer as compared to LiDAR, it is relatively invariant against external disturbances such as weather and lighting variations. RADAR information is utilized to train kNN by Raja Abdullah et al. (2016).



Figure 2.11: Experiment Setup for LiDAR-based VTR (Asborno et al., 2019)

In addition, Radio Frequency (RF) Transceiver is one of the viable options for vehicle recognition where it relies on the degree of RF signal perturbation to tell various vehicle types apart. Haferkamp et al. (2017) quantified the perturbation as the Received Signal Strength Indicator (RSSI) and used it to train SVM and kNN.

In Won et al. (2019), a WiFi-Transceiver-based vehicle recognition system is designed where spatial and temporal correlations of Wi-Fi channel state information (CSI), as well as phase data, are taken up by CNN to learn deep vehicle features.

In brief, the side-roadway methods have seen the utilization of acoustic sensors, LiDAR, RADAR, RF transceivers and Wi-Fi transceivers in characterizing the vehicles. Although they render poor performance under vehicle occlusion due to the installation point, they serve as a more economical choice as compared to on-roadway sensors. Table 2.2 summarizes the works discussed under the side-roadway category.

Table 2.2: Summary of Side-Roadway Methods

Reference	Sensor	Method	Dataset	#Classes	Metric	Highlight
Bischof et al. (2010)	Acoustic Sensor	Quadratic Discriminant Analysis	Private	2	77.5%	Features in time, spectral and cepstral domain are used to characterize the vehicles. The audio signal is used to perform image labeling for the training of an image-based classifier.
Asborno et al. (2019)	LiDAR	Naïve Bayes Ensemble	Private	5	81.0%	The combination of features from the upper and lower LiDAR units gives the best accuracy. Model performance can be improved by characterizing the vehicle shape.
Raja Abdullah et al. (2016)	RADAR	kNN	Private	3	99.0%	Distance between the receiver and vehicles is the determining factor for accuracy.
Haferkamp et al. (2017)	RF Transceiver	SVM	Private	2	99.1%	RSSI and vehicle length information achieve the best accuracy.
		kNN		2	99.6%	
Won et al. (2019)	Wi-Fi Transceiver	CNN	Private	5	91.1%	Based on the same input features, CNN performs better than SVM and kNN due to automated feature engineering.

2.4.1.3 Over-Roadway

Over-roadway is deemed as the best solution to overcome vehicle occlusion due to its high installation point above the ground. One of the sensors under the over-roadway category is the infrared ray sensor. It works based on ray reflection and it is less robust against ambient conditions. It is adopted in numerous studies (Khamayseh et al., 2015; Mei & Ling, 2011; Yang et al., 2016). Khamayseh et al. (2015) dedicated their work to differentiating between person and vehicle whereas Mei and Ling (2011) investigated VTR. On top of the infrared ray sensor, Odat et al. (2017) developed a hybrid system as illustrated in Figure 2.12. The system collects additional vehicle data from the ultrasonic sensor and the data is subsequently used for the training of the Bayesian network and neural network.



Figure 2.12: A Hybrid System for VTR (Odat et al., 2017)

The over-roadway sensors presented in this section are infrared ray sensors and ultrasonic sensors. They possess a complete view of the vehicles even under heavy traffic conditions and they are highly sought after for the application that requires a high percentage of vehicle coverage. Table 2.3 tabulates the works that employ the over-roadway sensors.

Table 2.3: Summary of Over-Roadway Methods

Reference	Sensor	Method	Dataset	#Classes	Metric	Highlight
Odat et al. (2017)	Infrared Ray & Ultrasonic Sensor	Bayesian Network	Private	5	99.8%	As errors in individual sensors are high, using multiple sensors improves the feature representation of the vehicle.
		Neural Network		5	77.7%	
Khamayseh et al. (2015)	Infrared Ray Sensor	Neural Network	Private	2	97.2%	Low scalability as the preprocessing step has to be customized for different use cases. Inference speed is 0.5 to 1 fps.
Mei and Ling (2011)	Infrared Ray Sensor	Static Template-based Classifier	Private	4	Unreported	The system performance can be examined further since it was tested on military vehicles which have large visual differences.

2.4.2 Computer Vision

2.4.2.1 Model-Based

The non-deformable 3D models are the early approaches from the model-based methods but they become irrelevant over time due to low flexibility. They are substituted by deformable car models which are highly versatile across various vehicles (Lin et al., 2014). Leotta and Mundy (2010) produced the deformable car model by employing 3D polygon mesh and 2D polygons to model the vehicle body, wheels and parts. After performing dimension reduction via PCA, the features are used to produce the hypothesized vehicle shape by fitting on 79 computer-aided-design (CAD) models. During inference, the hypothesized vehicle shape accommodates the actual vehicle by minimizing the edge distance through Gauss-Newton optimization before being classified using normal likelihood.

Krause et al. (2013) completed the 3D vehicle geometry estimation by matching the HOG features of a 2D test image to 3D CAD models via linear SVM. Thereafter, patch sampling via dart throwing technique (Cline et al., 2009) and patch rectification are carried out before applying RootSIFT (Arandjelović & Zisserman, 2012) for feature extraction. For a better pooling effect, Spatial Pyramid Matching (SPM) and BubbleBank (BB) are performed in 3D space and the pooled features are used to train linear SVM.

A landmark-based deformable car model is proposed by Lin et al. (2014). After predicting the vehicle landmarks, either HOG or Fisher Vector is used for vectorization. Subsequently, the distance between the 3D car model landmarks and predicted landmarks of the test vehicle image is minimized via pose and shape adjustment using landmark-based Jacobian. Their experiment demonstrates better classification performance when Fisher Vector is utilized during the vectorization process.

Ramnath et al. (2014) crafted the 3D vehicle curves by projecting 2D vehicle curves onto a reconstructed silhouette-based visual hull of the vehicle. To achieve viewpoint invariance, spurious edges of vehicles are eliminated and 3D models with relevant poses and positions are selected for Chamfer matching via Ellipse-based pose estimation. Then, the selected 3D vehicle curves are matched to the 2D points in the vehicle image. During the matching process, a transformation matrix is optimized by minimizing the Chamfer distance. Finally, the logistic regression ingests the average Chamfer distance and average orientation distance to infer the vehicle make and model (VMMR) information.

Dubská et al. (2014) presented a novel solution that estimates the 3D bounding box of a vehicle based on the 2D vehicle image. Firstly, the cameras are calibrated based on three types of vanishing points, which are those that coincide with traffic direction, perpendicular to traffic direction and perpendicular to the road surface. Thereafter, the 3D vehicle bounding box is generated according to the actual scale and it is used to identify the vehicle roof, front, rear, and side (Sochor et al., 2016) as depicted in Figure 2.13. The vehicle image then undergoes homography mapping and perspective warping before being taken up by CaffeNet (Jia et al., 2014) for classification. Sochor et al. (2016) disclosed that incorporating the vehicle orientation derived from the bounding box and the rasterized bounding box as the additional input features enhance the classification performance further.



Figure 2.13: Example of Original Vehicles (Top), 3D Bounding Box (Middle) and Unpacked Version (Bottom) (Sochor et al., 2016)

To summarize, the model-based methods were actively studied in the early times and numerous encouraging results were reported. Since the modeling of 3D vehicle models is intricate and the labeling process is daunting, falling back to 2D methods appears to be a wiser alternative (Sánchez et al., 2021). Consequently, an increasing number of research issues that steer the direction towards feature-based, novel backbone, unsupervised filter learning, part-based, attention and multi-scale features are seen. Table 2.4 shows the list of model-based methods reviewed in this section. They are evaluated either on private datasets, Stanford Cars (Krause et al., 2013) or Web-Nature Comprehensive Cars (CompCarsWeb) (Yang et al., 2015) datasets.

Table 2.4: Summary of Model-Based Methods

Reference	Method	Dataset	#Classes	Metric	Highlight
Leotta and Mundy (2010)	3D Vehicle Model based on 3D and 2D Polygons	Private	5	100%	Formulation of an accurate 3D model is challenging.
Krause et al. (2013)	3D Object Representation using 3D BB for Pooling	Stanford Cars	196	67.6%	3D models of the vehicles are required in advance.
Lin et al. (2014)	3D Model Fitting based on Vehicle Landmarks	Private	30	90.0%	The vehicle recognition performance is highly dependent on landmark prediction accuracy.
Ramnath et al. (2014)	3D Curve Alignment	Private	9	82.2%	3D models of the vehicles are required in advance.
Sochor et al. (2016)	CaffeNet based on 3D Bounding Box	CompCarsWeb	431	84.8%	A 3D bounding box can be predicted based on the 2D image.

2.4.2.2 Feature-Based

Feature-based methods observe the textures, gradient orientation and interest points to handcraft discriminative vehicle features. The features handcrafted upstream are normally channeled to ML algorithms to learn the decision boundary between vehicle classes. Sun et al. (2017) proposed a two-stage classifier for VTR. The global features, which are the vehicle edges, are detected using a hybrid technique powered by OTSU and the Canny operator to reduce the adverse effect of lighting variations. For local features, Gabor Wavelet Transform (GWT) is applied on the individual vehicle parts such as the vehicle roof, windscreen, rear-view mirror, hood and license plate instead of the full vehicle image to mitigate the effect of object occlusion. Eventually, kNN Probability Classifier (kNNPC) and Discriminative Sparse Representation-based Classification (DSRC) take in global and local features, respectively to infer the vehicle types. Despite exhibiting encouraging results, the two-stage classifier is less robust and time-consuming (Liao et al., 2022).

Tang et al. (2017) generated multi-scale and multi-orientation features by proposing the Local Gabor Binary Pattern Histogram Sequence (LGBPHS) that applies GWT and Local Binary Pattern (LBP) sequentially. The crafted features serve as input to a distance-based classifier. The solution is vulnerable under low light illumination (Xiang et al., 2019).

Instead of resorting to monoscopic cameras, Derrouz et al. (2019) built a framework based on stereoscopic vision. The stereoscopic vision allows them to extract the 3D vehicle parameters from the disparity maps. HOG is also included as one of the inputs to the classifier. Due to the utilization of a stereoscopic system, their solution is less economical.

In Manzoor et al. (2019), a comparative study between HOG and GIST is conducted and HOG is found to prevail due to higher accuracy and faster inference speed. Although HOG is widely used as a feature extractor (Derrouz et al., 2019; Krause et al., 2013; Lin et al., 2014), its inability to account for the spatial property of the object reduces the image perception capability (Zhang, 2012).

Since the handcrafted features are highly dependent on the raw pixel values, they impair the robustness of the system. The handcrafted features are vulnerable to environmental variations and this often lands the early works at a disadvantage (Chan et al., 2015; Dong et al., 2015; Ge et al., 2017; Huang et al., 2015; S. Li et al., 2018). To address this, several works put forward feature transformation techniques to encode the handcrafted features into mid-level features.

One of the well-known feature encoding schemes is Bag of Features (BoF). It quantizes the features by mapping them to the nearest visual words identified through k-Means clustering. Manzoor and Morgan (2017) paired SIFT with BoF where the SIFT keypoints are transformed into a frequency vector of visual words. As SIFT is utilized in the process, their solution is computationally slow (Huang et al., 2015; Siddiqui et al., 2016; Wen et al., 2015). Siddiqui et al. (2016) embarked on a more efficient approach by substituting SIFT with SURF. They also put forward a more flexible visual word-building scheme dubbed Modular-Dictionary (MD). The novel scheme maintains a set of class-specific visual words and full reconstruction can be evaded if an update is required. In general, aside from improving the feature robustness, BoF reduces the memory footprint as it reduces the high-dimensional keypoints to low-dimensional frequency vectors. However, this also implies the loss of information during the quantization process which causes performance degradation.

Nazemi et al. (2020) emphasized the locality constraint during the feature encoding process. The Locality-constraint Linear Coding (LLC) transforms the raw dense SIFT features and the resultant features are used to fit linear SVM. Compared to AlexNet (Krizhevsky et al., 2012), their solution renders stronger performance but it is limited to vehicle frontal images only.

To resolve BoF's information loss, Jamil et al. (2020) eliminated the quantization step in their Bag of Expressions (BoE). As portrayed in Figure 2.14, new expressions are generated by mean aggregating the visual words with the neighbors. Their technique is lauded for its high tolerance against occlusion and viewpoint variations and is also effective in addressing multiplicity and ambiguity problems. Since the utilization of kNN for neighbor identification imposes $O(MD)$ complexity where M is the number of visual words and D is the number of features, future work can look into increasing the computational efficiency.



Figure 2.14: Bag of Expressions (Jamil et al., 2020)

Owing to the feature encoding schemes, major improvements are seen for feature-based methods in terms of the solution robustness (Wang et al., 2010). However, the system performance is heavily dependent on the feature extractor algorithm and making the right choice requires substantial experience. The works proposed here also need to be validated further to examine their performances on the dataset with a higher number of classes (Krause et al., 2014).

To put it briefly, the feature-based methods that leverage raw features are generally susceptible to low-quality images. The advent of feature encoding schemes has successfully lifted the vulnerability to a certain extent but these solutions ought to be examined further on large-scale datasets that carry a vast number of classes. Table 2.5 summarizes the feature-based works that have been validated on Beijing Institute of Technology (BIT)-Vehicle (Dong et al., 2015), National Taiwan Ocean University-Make and Model Recognition (NTOU-MMR) (Hsieh et al., 2014) and Surveillance-Nature Comprehensive Cars (Yang et al., 2015) datasets. ‘*’ marks the best performance reported for the dataset.

Table 2.5: Summary of Feature-Based Methods

Reference	Method	Classifier	Dataset	#Classes	Metric	Highlight
Sun et al. (2017)	GWT & Canny Operator	Cascade Classifier	Sun et al. (2017)	4	93.0%	Characterize vehicle through patch-wise GWT and illumination invariant Canny operator. Less robust and time-consuming.
Tang et al. (2017)	LGBPH	Similarity Matching	Private	8	91.6%	Distance-based classification using LGBPH as features. Vulnerable to low lighting conditions.
Derrouz et al. (2019)	3D Parameters & HOG	SVM	BIT-Vehicle	6	95.2%	Use HOG features and 3D parameters of vehicles from stereo images as features. A stereo vision-based solution incurs higher hardware costs.
Manzoor et al. (2019)	HOG	SVM	NTOU-MMR	35	97.9%	HOG and linear SVM are used as feature extractor and classifier, respectively. HOG does not consider the spatial property of the object.
Manzoor and Morgan (2017)	SIFT & BoF	SVM	NTOU-MMR	37	89.0%	Encode SIFT features using BoF. SIFT is computationally slow.
Siddiqui et al. (2016)	SURF & BoF	SVM	NTOU-MMR	29	93.7%	MD builds class-wise visual words to provide flexibility in updating the dictionary
Nazemi et al. (2020)	Dense-SIFT & LLC	SVM, MLP	Private	50	97.5%	Adopt a fast non-linear feature encoding scheme (LLC) to encode SIFT features. Applicable to vehicle front images only.
			CompCarsSV	281	98.4%	
Jamil et al. (2020)	BRISK, HOG & BoE	SVM	NTOU-MMR	29	98.4%*	BoE aggregates the neighboring features to eliminate the vulnerability of BoF due to the viewpoint variations. It has $O(MD)$ complexity due to kNN.

2.4.2.3 Novel Backbones

Recent developments focus on ameliorating the CNN building blocks to elevate the feature representation learning as well as maintaining or even boosting the parameter efficiency. The advent of Vision Transformer (ViT) (Dosovitskiy et al., 2020) also spurs a separate development track using transformer architecture as the base that brings promising classification performance over a wide range of applications.

Iandola et al. (2016) invented a lightweight CNN called SqueezeNet. It is mainly composed of fire modules that perform squeezing and expanding operations using 1×1 and 3×3 convolutions. There is also a residual variant that implements skip connections to alleviate the vanishing gradient problem. Lee et al. (2019) experimented with SqueezeNet and Residual SqueezeNet and reported exceptional performance. Despite being 53 and 11 times smaller than AlexNet (Krizhevsky et al., 2012) and GoogLeNet (Szegedy et al., 2015), the computational efficiency needs to be improved if deployment to mobile devices is desired (Chen et al., 2020).

Aside from being lightweight, MobileNetv2 (Sandler et al., 2018) renders low latency by replacing the normal convolution with depth-wise and point-wise convolution. It is used by Boonsirisumpun and Surinta (2022) for VTR.

Ke and Zhang (2020) introduced a Dense Attention Network (DANet) for VMMR. The DA block adopts the dense connection strategy that densely connects all layers to encourage free information propagation. Although the dense connection avoids the vanishing gradient problem, it poses a risk of overfitting (Liu & Zeng, 2018). To pay attention to the salient feature responses, the DA block culminates with a Squeeze-and-Excitation (SE) block (Hu et al., 2018) that performs channel-based feature refinement.

Gholamalinejad and Khosravi (2021b) shared a superior pooling strategy in their Wavelet Deep Neural Network (WDNN). The pooling technique, which is 2D Discrete Wavelet Transform (DWT), uses Haar wavelet to perform low-pass and high-pass filtering. The experiment results reveal that DWT is incompatible with global average pooling (GAP) since it leads to poorer performance as compared to the primitive maximum and average pooling operations.

Constructing a CNN from scratch can be very daunting as there are no established rules or standards to guide the development course. Fortunately, Neural Architecture Search (NAS) (Elsken et al., 2019) and Differential Architecture Search (DARTS) (H. Liu et al., 2018) are the succors to the pain. They dismiss the human trial and error process by employing optimization techniques to deduce a competitive architecture. The arising concerns from these methods are they have expensive search costs and there is no way to include the custom-designed module. As a workaround, Tanveer et al. (2021) demonstrated in their fine-tuning DARTS strategy that the selected module i.e. SE module can be considered during the optimization process. Although their methodology provides great flexibility in network design, the justification for choosing the SE module is not provided and the performance can be adversely affected.

Several alterations are made to EfficientNet (Tan & Le, 2019) to produce EfficientNetv2 (Tan & Le, 2021) which possesses lower computational cost in terms of training duration and parameter efficiency. In particular, the latency is reduced by substituting the MBConv blocks at the early layers with Fused-MBConv blocks to enable the execution of regular convolution that is modern accelerator-friendly. For the scaling of the EfficientNetv2 architecture, a non-uniform scaling strategy is opted so that layers are scaled selectively with the consideration of training efficiency. For an effective

learning process, the training also adopts the progressive learning strategy to tune the regularization strength along with the image resolution.

Similar to EfficientNetv2 (Tan & Le, 2021), TResNet (Ridnik et al., 2021a) evolved from its predecessor which is ResNet (He et al., 2016) to maintain the training and inference efficiency. One of the 5 refinements is the replacement of downsampling operations at the early layers with the lightweight SpaceToDepth layer to avoid aggressive information loss. To enhance translation-equivariance, the anti-alias downsampling layer substitutes the 3×3 , stride 2 convolution layer with the 3×3 , stride 1 convolution layer and a fixed blur filter. Moreover, the Inplace-Activated BN layer is opted to achieve memory footprint reduction for the enablement of larger training batch size. For better speed-accuracy trade-off, a hybrid building block made up of BasicBlock and Bottleneck block is adopted in TResNet architecture. The location of the SE block is also revised to maximize the gain in speed and accuracy and its usage is limited to the first three convolutional blocks.

Given the limited receptive field size of convolution kernels, the ability of CNN to learn global features is compromised. Wu et al. (2022) built a Multi-Granularity Feature Learning (MGFL) Network that is free from the shackles of convolution kernels with the help of the Graph Neural Network (GNN). Rather than connecting the feature responses in structured ways, GNN is formless and the connection is highly customizable due to the existence of nodes and edges in the graph structure. In their proposal, an abstract graph branch constructs a graph RNN to perform information exchange among image pyramids. In the detailed graph branch, the Graph Convolutional Network (GCN) exploits the semantic relationship between cross-hierarchy labels and regulates the feature space to render more cohesive class-wise embeddings. It is worth noting that the annotation

process for MGFL is intensive as the labels at different granularity levels are required by the detailed graph branch to model multi-hierarchy information.

ViT (Dosovitskiy et al., 2020) can be considered the forefather of the transformer architecture in the CV domain. It first converts the image into a 1-dimensional vector, which is a sequence of image tokens. To prevent permutation-equivariance, positional information is added to the image tokens and they are passed on to the encoder layers. Within the encoder layer, the MHSA computes the attention matrix by quantifying the correlation of query and key matrices in different subspaces and the attention matrix is subsequently used to weigh the value matrix to increase the attention given to the significant image tokens. After MHSA, the FFN, which is a MLP, performs the non-linear transformation. Along the layers, the learnable class token aggregates information from all image tokens and it is eventually used as the input to the classification head. Serving as the base for the development of future transformer architecture, it is criticized for its moderate capability in encapsulating the information carried by all image tokens (Touvron et al., 2021; Yuan et al., 2021).

Just like CNN, the transformer also faces the issue of saturated performance with increasing depth. In Class-Attention in Image Transformers (CAiT) (Touvron et al., 2021), the contribution of MHSA and FFN components is adaptively added with the help of LayerScale, a channel-wise multiplication vector. The Multi-Head Class Attention (MHCA) is also added as the last encoder layer to enhance the ability of class token in information encapsulation. CAiT has a higher computational cost but it achieves better performance as compared to ViT.

The lack of inductive bias often requires the transformer architecture to be trained on a huge amount of training data to achieve exceptional performance. Convolution-enhanced image Transformer (CeiT) (Yuan et al., 2021) overcomes this with a transformer

architecture partly powered by convolution operations to retain the inductive bias and to strengthen the locality. To preserve the spatial relationship of the pixels, the Image-to-Tokens (I2T) layer uses convolution for tokenization. Locally-Enhanced FFN (LEFFN) reinforces the local connectivity of the image tokens by employing 2D depth-wise convolution for non-linear transformation. Since the class token is relatively weaker in summarizing the learned information, a Layer-Wise Class Token Attention (LWCTA) mechanism is introduced to process the class token embeddings from all layers with MHSA and FFN to comprehend the multi-granularity information. Despite the success in inducing the spatial inductive bias, CeiT is suboptimal as it disregards the underlying object structure (Kang et al., 2022).

LeViT (Graham et al., 2021) is a convolution-like high-speed transformer. It follows the design convention of CNN by downsampling the spatial resolutions and increasing the number of channels with increasing depth. To maintain the intensity of positional information, attention bias is repeatedly injected at every MHSA layer. Instead of relying on the class token, average pooling is used to compile the learned information. It is reckoned that the sacrifice of accuracy is traded for speed improvement as the image pyramids bring information loss (Xu et al., 2021).

MHSA is credited for being able to track long-range dependencies but it is plagued by the quadratic complexity. In the Global Filter Network (GFN) (Rao et al., 2021b), the global attention computation is reduced to log-linear complexity in the frequency domain. Upon converting the image tokens into the frequency domain with 2D Discrete Fourier Transform (DFT), global filters are applied to uncover the new features. Subsequently, the features are transformed back to the spatial domain before being picked up by the FFN. The global filters are considered a decent replacement for MHSA since it is a depth-

wise global circular convolution in disguise, which can capture both long-term and short-term interaction.

Touvron et al. (2022) devised a transformer architecture that runs MHSA and FFN in parallel to reduce computational complexity. Although parallel operation eases the optimization process, significant differences can only be seen in the deep models i.e. ViT-L. Moreover, their experiment discloses that fine-tuning MHSA alone produces comparable results as compared to full fine-tuning. The technique is recommended for transfer learning for tasks within similar domains.

The ViT performance is subpar for fine-grained classification, especially VMMR, as it is designed for general image classification tasks. Using ViT as the base, He et al. (2021) introduced a Fine-Grained Transformer (TransFG) as shown in Figure 2.15. Specifically, a Part Selection Module (PSM) is added after the last encoder layer and it bases the selection of discriminative tokens on the attention matrix of MHSA. Doing so allows the classification head to study the class-specific features effectively without being carried away by the irrelevant tokens. Although the results demonstrate a 1.1% improvement over ViT on the Stanford Cars dataset (Krause et al., 2013), the approach is accompanied by the risk of overfitting.

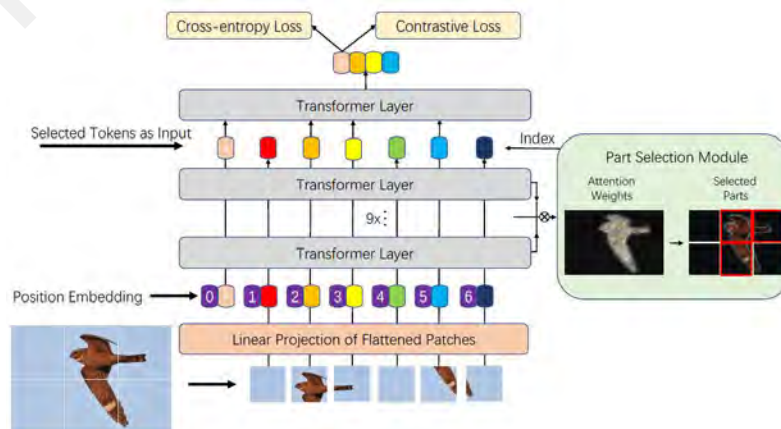


Figure 2.15: Fine-Grained Transformer (He et al., 2021)

Echoing He et al. (2021), H. Liu et al. (2023) spotted the relevant image tokens via the Feature in Feature Abstraction (FFA) module in their Invariant Cues-Aware Feature Concentration Transformer (TransIFC). Rather than quantifying the token relevancy against the class token alone, a pairwise token-to-token similarity is considered for the selection of discriminative tokens. The identified tokens are then processed by the final encoder layer to refine the feature representation. Additionally, the Hierarchy Stage Feature Aggregation (HSFA) module ingests feature embeddings from all encoder layers to synthesize multi-scale features in computing classification logits. It is construed that FFA precludes diversified learning by limiting the information source from image tokens with high similarity. For HFSA, involving extremely low-level information in logit computation brings disruptive and noisy signals and this affects the representation capacity of the network.

In essence, the novel backbones category presents a diverse number of vehicle recognition solutions that are built upon CNN, GCN or transformers. Nevertheless, most of them still lack the fine-grained feature extraction ability which is believed to be able to propel the performance further if being incorporated. Table 2.6 summarizes the works based on novel CNN and transformer architectures. The benchmark datasets include Vehicle Type Image Data (VTID) (Boonsirisumpun & Surinta, 2022), Iranian Vehicle Dataset (IRVD) (Gholamalinejad & Khosravi, 2021a), Miovision Traffic Camera Dataset (MIO-TCD) (Luo et al., 2018), Car-FG3K (Wu et al., 2022), CompCarsWeb, CompCarsSV and Stanford Cars.

Table 2.6: Summary of Novel Backbones

Reference	Method	Dataset	#Classes	Metric	Highlight
Lee et al. (2019)	Residual SqueezeNet	Private	766	96.3%	Apply lightweight CNN for VMMR. The model size is still large for mobile devices.
Boonsirisumpun and Surinta (2022)	MobileNetv2	VTID	5	93.4%	Use MobileNetv2 for VTR. It performs poorer than MobileNet.
Ke and Zhang (2020)	DANet	Stanford Cars	196	mAP 94.5%	Use DenseNet-like architecture as CNN building blocks. Prone to overfitting due to dense connection.
Gholamalnejad and Khosravi (2021b)	WDNN	IRVD	5	99.1%	Introduce DWT, a Haar transform-based pooling strategy. DWT increases training time due to added gradient computation during backpropagation.
		MIO-TCD	11	95.1%	
Tanveer et al. (2021)	Fine-tuning DARTS	CompCarsWeb	431	95.9%	Fine-tuning DARTS considers manually designed architectures during the search for novel architectures. The manually designed architectures to include during the search is subjective to individual judgment.
		CompCarsSV	281	99.2%*	
Tan and Le (2021)	EfficientNetv2-L	Stanford Cars	196	95.1%	Revision of EfficientNet architecture to become lightweight and more accurate.
Ridnik et al. (2021a)	TResNet-L	Stanford Cars	196	96.0%	Revision of ResNet architecture to attain better speed-accuracy trade-off.
Wu et al. (2022)	MGFL	Car-FG3K	1,892	87.5%	Exploit the relationship between hierarchies using a graph neural network. Require annotations from coarse-to-fine level.
Dosovitskiy et al. (2020)	ViT-B	Stanford Cars	196	93.7%	First transformer architecture in CV domain. The image tokenization technique causes information lost. Class token fails to summarize patch information holistically.
Touvron et al. (2021)	CAiT	Stanford Cars	196	94.2%	MHCA is inserted after the last encoder layer to summarize patch information. MHCA adds slight computational costs to the original ViT.
Yuan et al. (2021)	CeiT-S	Stanford Cars	196	94.1%	Tokenize image through I2T, enforce spatial relationship among tokens through LE FFN and encapsulate cross-layer information through LWCTA. Object structures are disregarded.

Table 2.6: Summary of Novel Backbones, continued

Reference	Method	Dataset	#Classes	Metric	Highlight
Graham et al. (2021)	LeViT-192	Stanford Cars	196	89.8%	Produce pyramidal feature maps like CNN. Attention bias is repeatedly inserted into MHSA at all levels to provide positional information and to encourage flip invariance property. Deep narrow pyramidal structure causes information loss.
Rao et al. (2021b)	GFNet-H-B	Stanford Cars	196	93.2%	Attention among tokens is computed in the frequency domain using DFT. Has log-linear complexity.
Touvron et al. (2022)	ViT-L	Stanford Cars	196	93.8%	Execute MHSA and FFN in parallel to achieve speed-up. Fine-tuning MHSA is only feasible for transfer learning to tasks from similar domains.
He et al. (2021)	TransFG	Stanford Cars	196	94.8%	Transformer for fine-grained classification where PSM selects only significant patches for classification. PSM has an overfitting issue.
H. Liu et al. (2023)	TransIFC	Stanford Cars	196	94.7%	Discriminative tokens identified via FFA are used to infer class probability. FFA prevents diversified learning.

2.4.2.4 Unsupervised Filter Learning

Unsupervised filter learning refers to a group of works that learns convolution kernels without using the annotations. The convolution kernels are normally optimized beforehand by unsupervised algorithms such as PCA. They are then used to convolve with the images and the resultant features are used to fit the classifier.

Dong et al. (2015) proposed Sparse Laplacian Filter Learning (SLFL) to optimize the convolution kernels. SLFL deduces superior convolution kernels by minimizing the dictionary reconstruction error, optimizing the sparsity in feature distributions, projecting the data points based on manifold assumption and reducing the error between data points and their corresponding sparse representations. Apart from that, multi-task learning is implemented to optimize the softmax classifier using Kullback-Leibler (KL) divergence loss. Feature fusion technique is also practiced here where the input to the softmax classifier is the feature embeddings consolidated from the first and second stages of convolution operations. The analyzed results suggest that the network is still lacking in discriminating between Sport Utility Vehicles (SUVs) and sedans.

The principles of local connectivity and parameter sharing cause CNNs susceptible to rotation and scaling. Gao and Lee (2016) addressed this concern with Local Tiled CNN (LTCNN), which is an extension of TCNN (Ngiam et al., 2010). In pursuit of more distinguished features, LTCNN constraints the parameter sharing within the local sub-regions as illustrated in Figure 2.16. Each local sub-region learns the convolution kernels via Topographic Independent Component Analysis (TICA) (Hyvärinen & Hoyer, 2001) which associates the dependence of components with the proximity. As part of their VMMR pipeline, the symmetrical filter first detects the vehicle front and it is then converted into the HOG image. Thereafter, LTCNN is applied to the HOG image and the resultant feature vectors serve as input to the classifier. The solution delivered 98.5%

accuracy for 107 vehicle models, outperforming AlexNet (Krizhevsky et al., 2012) by 10%.

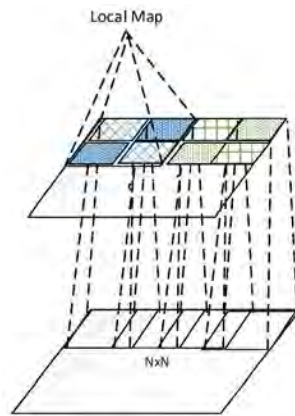


Figure 2.16: Local Tiled CNN (Gao & Lee, 2016)

Huang et al. (2015) learned the convolution kernels via the notable PCA in their PCA-based CNN. Essentially, the learned kernels are the eigenvectors of the top few principal components deduced from the image patches. It is also reported that the PCA filters are discriminative enough to be used directly for convolution and this makes the fine-tuning of PCA filters unnecessary. Consequently, the PCA-CNN has a stark training duration difference with backpropagation-optimized CNN, which is 15 minutes against 15 hours. What is more encouraging is the short training duration also results in higher accuracy i.e. 99.1%, prevails over the normal CNN by 1% accuracy.

A similar strategy is adopted by Soon et al. (2018) and Soon et al. (2020). The PCA-based CNN implements two PCA-based convolutions on vehicle headlamps and full vehicle bodies for VMMR and VTR, respectively. Despite reporting high accuracy, having a short inference time and being highly robust Chan et al. (2015) and Fang et al. (2016) conjectured that PCA filters cause the CNN to underperform as they refrain the CNN from learning fine-grained features.

In summary, the discussed unsupervised filter learning techniques cover SLFL, TICA and PCA. They act as a drop-in replacement for the computationally expensive backpropagation and are notable for the ability to optimize the convolution kernels without the need for ground-truth labels. On the flip side, this signifies the optimization process does not account for the maximization of class separability and it leads to an underperforming CNN. Table 2.7 tabulates the unsupervised filter learning-based vehicle recognition literature.

Universiti Malaya

Table 2.7: Summary of Unsupervised Filter Learning

Reference	Method	Dataset	#Classes	Metric	Highlight
Dong et al. (2015)	SLFL & Multitask Learning	BIT-Vehicle	6	88.1%	The system fails mostly in differentiating SUVs and sedans.
Gao and Lee (2016)	HOG-LTCNN & Linear SVM	Private	107	98.5%	The system is tested with frontal images only.
Huang et al. (2015)	PCA-Based CNN	Huang et al. (2015)	10	99.1%	Convolution kernels are learned through PCA. Incremental PCA is needed to compute the eigenvectors when all training images cannot fit into the memory in one go.
Soon et al. (2018, 2020)	PCA-based CNN	CompCarsSV BIT-Vehicle	397 6	89.8% 88.5%	

2.4.2.5 Part-Based

Faster R-CNN (Ren et al., 2015) is one of the most representational works for supervised part-based methods. It generates region proposals via the Region Proposal Network (RPN) and pools the Region of Interest (ROI) into fixed-size vectors via ROI pooling for class and bounding box predictions. More importantly, it reports significant speed improvements from its predecessors, which are R-CNN (Girshick et al., 2014) and Fast R-CNN (Girshick, 2015). Arinaldi et al. (2018) and X. Wang et al. (2019) built VTR solutions based on Faster R-CNN but their solutions are still less favorable when it comes to real-time inference.

Single Shot Detector (SSD) (W. Liu et al., 2016) is a one-stage object detector that renders high inference speed with decent performance. The ingestion of multi-scale feature maps from the image pyramids empowers the network in detecting both large and tiny objects. Satar and Dirik (2018) experimented with VGG-based SSD for vehicle detection and the classification task was delegated to ResNet (He et al., 2016). They proved that such a combination prevails over SSD alone.

You Only Look Once (YOLO) (Redmon et al., 2016) is another popular one-stage object detection algorithm. It is constructed upon DarkNet (Redmon & Farhadi, 2017, 2018) and is consistently refined over the years (Bochkovskiy et al., 2020; Redmon & Farhadi, 2017, 2018). Yang et al. (2019) dedicated their studies to VLR using YOLO architecture. They include multiple 3×3 and 1×1 convolution kernels into DarkNet (Redmon & Farhadi, 2017, 2018) as well as an upsampling layer following the last convolution layer. The proposed changes are necessary for the detection of tiny objects under the low vehicle logo-to-image ratio. The modified DarkNet produces better detection than YOLOv2 (Redmon & Farhadi, 2017) and future work should look at reducing the error rate of detecting letter-like logos.

Tajar et al. (2021) performed pruning on Tiny-YOLOv3 (Adarsh et al., 2020) to make it lightweight. Pruning results in the removal of BN and consecutive convolution layers. The pruned network is 1/3 smaller and 1.4 times faster at the expense of losing 0.5% mAP.

Zhao et al. (2022) presented YOLOv4-AF by revisiting the Path Aggregation Network (PAN) (S. Liu et al., 2018). Additional side linkages are established between the high-level feature maps to improve the flow of information and enhance the ability to detect tiny objects.

Under the weakly supervised regime, Besbes et al. (2020) proposed a dynamic vehicle parts selection strategy that works for multi-view scenarios. They reason that viewpoint variations cause certain vehicle parts to be hidden and defining a fixed set of vehicle parts to be used is unfeasible. Therefore, a look-up table is developed to guide the selection of the most discriminative combination of vehicle parts. After detecting the parts using YOLOv3 (Redmon & Farhadi, 2018), each part combination is sent to a VGG16-based Multi-Stream Network (MSN) and the dynamic fusion layer computes the logits based on parts availability. The solution requires huge model maintenance effort since the number of MSNs to be trained is equivalent to the number of part combinations stored in the look-up table.

Multi-Attention CNN (MA-CNN) (Zheng et al., 2017) pinpoints the discriminative vehicle parts in a weakly supervised manner. Within MA-CNN, the channel grouping and weighting layer perform channel grouping for vehicle parts identification and these parts are subsequently classified by the part classification network. On top of the well-known cross entropy loss, the network is optimized by channel grouping loss to encourage compact and diverse part learning. The downside of MA-CNN is the training pipeline is

complicated as the training should fire one of the loss functions alternately for convergence purposes.

Weakly-Supervised Data Augmentation Network (WS-DAN) (Hu et al., 2019) suggests a part localization technique guided by the attention maps. The attention maps, after being binarized, are used to produce binary cropping and dropping masks that highlight the prominent parts and diversify the attention regions, respectively. To produce richer embeddings, the second-order statistics of convolutional feature maps and attention maps are deduced via Bilinear Pooling (BP). Center loss (Wen et al., 2016) is also incorporated into the loss function to improve the cohesiveness of learned feature embeddings. The training of WS-DAN is memory intensive as the attention cropping and dropping operations double the number of images to be processed.

To eliminate multiple forward passes for the learning of localization, Efficient Localization, Pooling and Embedding Network (ELoPE) (Hanselmann & Ney, 2020a) prepends a lightweight localization module to the training pipeline. The localization module possesses strong semantic understandings owing to the Knowledge Distillation (KD) from deep layers through l_1 loss. The network first puts the image through the localization module before pushing the segmented vehicle to ResNet for feature extraction. Furthermore, global k -max pooling retrieves the top- k activation values for better information encapsulation. Metric learning is also implemented to regularize the feature embeddings in the feature space. Despite being lightweight and more computationally efficient, the localization module performs beneath expectations under multi-object circumstances. Multiple training runs are needed to optimize the localization and classifier network too (Hanselmann & Ney, 2020b).

In the Mixture-of-Expert-Attention Swapping Network (ME-ASNet) (L. Zhang et al., 2021), the swapping of prominent parts between an image pair is implemented as a data

augmentation technique and a feature maps binarization step similar to Hu et al. (2019) and Hanselmann and Ney (2020a) is used for salient object detection. Both GAP and Global Maximum Pooling are inserted after the top-level feature maps to pool both the average and salient feature responses. ME-ASNet is optimized by cross entropy loss using raw and cropped images as inputs.

Multi-Branch and Multi-Scale Learning Network (MMALNet) (F. Zhang et al., 2021) achieves better localization performance than Hu et al. (2019) and Hanselmann and Ney (2020a) by generating the object masks from the last two convolution layers in Attention Object Location Module (AOLM). Besides, instead of opting for feature maps binarization and minimum bounding rectangle techniques for parts slicing, they use non-maximal suppression (NMS) with the part significance quantified as the average feature responses within the window. This promotes diversified learning by ensuring no identical parts are selected. Since the proposed strategy is limited to training, it adds no computational burden to the solution during inference.

Spatial Transformer Network (STN) (Jaderberg et al., 2015) is another viable option to realize vehicle localization. By pretraining the STN, it can predict the affine transformation matrix to segment the significant vehicle parts. The segmented vehicle enables a more accurate prediction by Wide ResNet50 (Zagoruyko & Komodakis, 2016) downstream. The framework dubbed Multiscale Attention Windows Network (MAWNet) (Ghassemi et al., 2019) shows exceptional performance but the training process is less straightforward. STN and WideResNet50 are not trainable end-to-end. The pretraining of WideResNet50 also requires bounding box annotation.

Contrastively-Reinforced Attention CNN (CRA-CNN) (Liu et al., 2020) adopts a compare-and-contrast strategy in learning fine-grained visual cues. The vehicle and background scene are segregated via the affine transformation matrix predicted by STN.

A constructive reinforcement loss guides the representation learning process by maximizing the distance of deep feature embeddings of key and redundant vehicle parts. It is worth noting that the top-level information is accessed directly for effective learning of the affine transformation and hence KD is not required in their work.

Hanselmann and Ney (2020b) enhanced their previous work (Hanselmann & Ney, 2020a) with Attention Network (AttNet) and STN-based Affine Network (AffNet) as shown in Figure 2.17. The practice of learning the affine parameters from low-level feature maps is maintained to avoid multiple forward passes. The semantic knowledge of the top-level feature maps is passed on to the AffNet via $l1$ loss. The uniqueness of their implementation is the horizontal and vertical transformation parameters are predicted separately using the maximum pooled row and column vectors from AttNet’s attention maps. Their proposal has seen a remarkable improvement in positioning performance. The classification performance is also superior to Ghassemi et al. (2019) and Liu et al. (2020) despite using the semantically weak shallow-layer feature maps for localization.

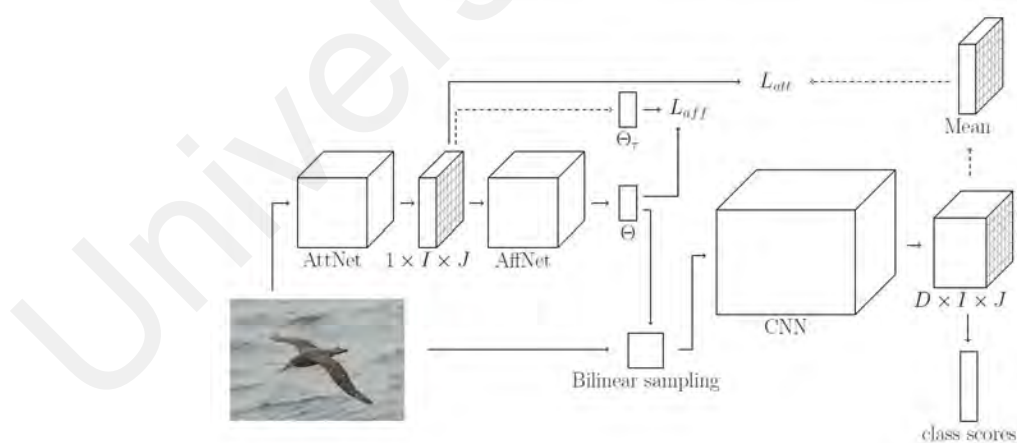


Figure 2.17: End-to-end Localization and Classification Model (Hanselmann & Ney, 2020b)

The need for multiple forward passes is inevitable for most weakly supervised techniques (Hu et al., 2019; Liu et al., 2020; F. Zhang et al., 2021; L. Zhang et al., 2021),

be it those rely on feature maps binarization or STN for localization. The KD technique presented by Hanselmann and Ney (2020a) and Hanselmann and Ney (2020b) can be the workaround considering the exceptional localization and classification abilities even with one forward pass.

Zhu et al. (2021) utilized OTSU for vehicle segmentation. The high-level feature maps are also exploited for the learning of pixel-level masks to facilitate the identification of fine-grained vehicle details. To produce better prediction, combined logits of the raw image, segmented vehicle and pixel-level masks are utilized to augment the explanatory power of the network.

Rachmadi et al. (2018) proposed a primitive part-based technique that slices the image into 4 patches of equal sizes and each of them utilizes a ResNet18 (He et al., 2016) for deep feature extraction. The deep feature embeddings of the patches as well as the raw image are received by LSTM to model the inter-part relationship. Their solution should be made more efficient by considering a shared ResNet18 for feature extraction purposes to reduce the computational cost significantly.

Feature Fusion Car Model Classification Network (FF-CMNet) (Yu et al., 2018) adopts an approach akin to Rachmadi et al. (2018) but with fewer image partitions. The UpNet and DownNet are responsible for describing the upper and lower parts of the vehicle image and the feature vectors are eventually combined via FusionNet to form the classification logits. Generally, the part-based techniques practiced by Rachmadi et al. (2018) and Yu et al. (2018) are straightforward and hassle-free but the spatial structure of the vehicle is not taken into account. Slicing the object at the wrong position risks the loss of contextual information and this may hinder the learning process.

Siamese Self-Supervised Learning Network (SSLNet) (Ji et al., 2023) performs attention cropping and dropping based on feature maps binarization. The generated cropped image corresponds to the localized vehicle whereas the dropped image has the insignificant image regions nullified. Both of them are essential in rendering a focused learning process for the network. Additionally, the l_2 distance of both types of images is minimized for the learning of view-invariant feature embeddings. BP is used for high-quality feature encapsulation but the resultant huge feature dimension causes the network to be computationally expensive (Wei et al., 2021).

A. Li et al. (2022) are motivated to mitigate the impact of object occlusion on classification performance. In their Global Information-Assisted Network (GIAN), the legitimacy of vehicle parts obtained from MA-CNN (Zheng et al., 2017) is validated using the global information extracted from Global Attention-Concentrated CNN (GAC-CNN). GAC-CNN is essentially a GCN that is empowered to track long-range dependencies. However, unlike the vanilla MHSA that employs no masking during the attention matrix calculation, a spatial constraint is imposed to limit the attention view to the concentrated and criss-cross positions. This elevates the proportion of local information in global information encoding and allows the fine-grained vehicle features to be valued more. Based on the global information, cosine similarity between the feature centers and detected vehicle parts is employed as a metric to segregate the discriminative and non-discriminative parts. Thereafter, the discriminative parts are fused with the global features to improve the feature perception ability whereas the non-discriminative parts are described by global features fully to suppress the repercussion of object occlusion. To ease the transfer of correctly learned information, KL divergence loss is employed for KD between the global and local features. At the expense of high computation and memory costs, the performance of GIAN is indeed encouraging.

Teacher-Student-Based Attention CNN (T-S-ACNN) (A. Li et al., 2023) is an improvisation over GIAN (A. Li et al., 2022) in terms of both classification performance and parameter efficiency. The incorporation of boundary loss into MA-CNN (Zheng et al., 2017) contributes to noise removal beyond the object area and more superior localization power is seen. Similarly, GCN which exerts a global receptive field is adopted for unobstructed interaction among the feature responses selected by the local attention masks from MA-CNN. Aside from cross entropy loss, KL divergence loss is used to implement collaborative learning to promote the learning of relevant features. Domain adaption can be considered in the future to improve the network performance.

All in all, the part-based methods unify the learning of localization and classification tasks in a single framework. Some of them even acquire localization ability through weakly supervised manners by exploiting the feature responses instead of the bounding box information. However, the issue of multiple forward passes and the convergence problem of loss function have prevented these solutions from dominating the vehicle recognition domain. Table 2.8 summarizes the part-based literature. Aside from BIT-Vehicle, CompCarsWeb, Stanford Cars, MIO-TCD, CompCarsSV, additional datasets being used are Pascal VOC (Everingham et al., 2010), MIT Traffic (Wang et al., 2008) and Vehicle Logo Detection (VLD)-30 (Yang et al., 2019).

Table 2.8: Summary of Part-Based Methods

Reference	Method	Backbone	Dataset	#Classes	Metric	Highlight
Arinaldi et al. (2018); X. Wang et al. (2019)	Faster R-CNN	-	Pascal VOC	4	mAP 81.1%	Use Faster R-CNN for VTR. Barely meet the need for real-time inferencing.
		-	MIT Traffic	6	mAP 69.4%	
Satar and Dirik (2018)	SSD & ResNet	-	Private	7	95.1%	Employ SSD and RestNet for detection and classification, respectively. Not suited for lightweight devices.
Yang et al. (2019)	Modified YOLO	DarkNet	VLD-30	30	mAP 89.9%	Amend DarkNet for VLR. The revised architecture fails on letter-like vehicle logos.
Tajar et al. (2021)	Trimmed Tiny-YOLOv3	DarkNet	BIT-Vehicle	6	mAP 95.1%	Prune YOLOv3-tiny to achieve speed-up. Perform slightly worse than YOLOv3-tiny.
Zhao et al. (2022)	YOLOv4-AF	DarkNet	BIT-Vehicle	6	mAP 83.5%	PAN is expanded to three routes to aid tiny object detection. PAN increases computational costs.
Besbes et al. (2020)	Multi-Stream Network	VGG16	CompCarsWeb	431	95.1%	Dynamic fusion layer fuses vehicle parts based on parts availability. The number of convolutional trunks is linear to the number of part combinations.
Zheng et al. (2017)	MA-CNN	VGG19	Stanford Cars	196	92.8%	Perform channel clustering. It is trained using channel grouping loss. It has a complex training strategy as cross entropy and channel grouping loss are trained alternately.
Hu et al. (2019)	WS-DAN	Inception-v3	Stanford Cars	196	94.5%	Attention-guided cropping crops meaningful parts of the image during training and randomly drops object parts to prevent overfitting. Require two forward passes and the input for the second pass is twice the amount of the first pass.
Hanselmann and Ney (2020a)	ELoPE	ResNet101	Stanford Cars	196	95.0%	Localize objects by binarizing feature maps, pool top salient features through K-max pooling, and adopt metric learning to optimize the feature space of embeddings. The localization module requires multiple separate training runs.
L. Zhang et al. (2021)	ME-ASNet	ResNet50	Stanford Cars	196	94.8%	AS swaps discriminative regions between images to prevent overfitting whereas ME localizes objects and optimizes networks in multiple passes.

Table 2.8: Summary of Part-Based Methods, continued

Reference	Method	Backbone	Dataset	#Classes	Metric	Highlight
F. Zhang et al. (2021)	MMALNet	ResNet50	Stanford Cars	196	95.0%	AOLM binarizes feature maps and APPM ensures diversification among the chosen discriminative parts via NMS. Sliding window in APPM is time-consuming.
Ghassemi et al. (2019)	MAWNet	Wide ResNet50	Stanford Cars	196	94.8%	Apply ResNet18-based STN to segment discriminative vehicle parts and Wide ResNet50 as the classifier. STN and CNN are not trainable end-to-end. Pre-training Wide ResNet50 requires bounding box annotation.
			CompCarsWeb	431	97.8%	
Liu et al. (2020)	CRA-CNN	ResNet101	Stanford Cars	196	94.8%	Localize the vehicle using a STN-like module and elevate the uniqueness of key features through contrastive reinforcement loss. STN has a convergence problem. Require two forward passes.
Hanselmann and Ney (2020b)	AttNet & AffNet	ResNet101	Stanford Cars	196	95.6%	STN-like AttNet and AffNet learn affine transformation guided by high-level feature maps. STN is hard to converge.
Zhu et al. (2021)	OTSU-based Segmentation & Multi-Loss	ResNet101	Stanford Cars	196	95.0%	Use image-level, object-level, pixel-level loss and feature fusion to supervise the learning. OTSU-based segmentation is less robust.
Rachmadi et al. (2018)	SPM-LSTM-CNN	ResNet18	MIO-TCD	11	98.0%*	Model part-to-part relationship via LSTM. The number of CNNs used is linear to the number of image partitions.
Yu et al. (2018)	FF-CMNet	-	CompCarsSV	281	98.9%	Perform feature fusion after processing the upper and lower part of the vehicles independently. Less robust as the image partitioning strategy assumes the vehicle takes the central position of the image.
Ji et al. (2023)	SSLNet	ResNet101	Stanford Cars	196	95.5%	Object cropping and part erasing are carried out based on part-based feature maps. High dimensional vectors obtained from BP are used to produce logits.
A. Li et al. (2022)	GIAN	ResNet50	Stanford Cars	196	95.7%	Local and global features are synergized to enrich features and mitigate object occlusion. The network is bulky.
A. Li et al. (2023)	T-S-ACNN	ResNet152	Stanford Cars	196	95.9%	Ameliorate MA-CNN to improve parts extraction and the interdependencies among part-level features are learned through GCN.

2.4.2.6 Attention

The attention domain is an immense field of study inspired by human vision. Different from part-based approaches, it downweighs the activation values of the insignificant regions instead of alienating them completely. It ensures these regions exercise minimal influence towards the representation learning process whilst keeping the contextual information of an image intact.

Elkerdawy et al. (2018) implemented co-occurrence learning to exploit the spatial dependency among the top-level feature maps with a co-occurrence layer. Specifically, the co-occurrence matrix encodes the relationship of the parts by computing the correlation of feature maps. To ensure modest computational complexity, a channel reduction operation is carried out before the co-occurrence layer but it endangers the learning process due to information loss (Forcen et al., 2020).

In the Attribute-Aware Attention Model (A3M) (Han et al., 2018), an attribute-category reciprocal attention module is introduced to determine the intrinsic attributes that serve best as the category-specific features. Based on the category and attribute embeddings extracted from the backbone networks, the attribute-guided attention and category-guided attention perform dot-product operations for regional and attribute feature selection, respectively.

A compare and contrast strategy is adopted in the Attentive Pairwise Interaction Network (APINet) (Zhuang et al., 2020). After encapsulating the crucial features of an image pair in the mutual vectors learning stage, the contrasting process is carried out by a gating mechanism to highlight the semantic differences, which are also the unique attributes of both objects. These unique attributes are then used to complement the image vectors to produce attentive feature vectors to tell the significant regions apart from the rest. Both cross entropy and hinge loss are used to quantify the learning quality. The

performance of APINet is highly dependent on the image pairs. It has a relatively lower generalization ability due to the need to customize a dataset-specific sampling strategy.

Channel Interaction Network (CIN) (Gao et al., 2020) features Self-Channel Interaction (SCI) and Contrastive Channel Interaction (CCI) for effective channel communications. To promote diversified learning, SCI performs the bilinear operation to identify the negatively correlated information as a means to uncover the semantically complementary channel information. CCI involves cross-image interaction where it preserves the prominent channel information and discards the mutual information to reduce the noise level through subtraction between SCI matrices. The training process of CIN is guided by cross entropy and contrastive loss. It also requires a careful image pair construction strategy to ensure convergence of loss function (T. Zhang et al., 2021).

The Convolutional Attention Model (ConvAM) (Yu et al., 2020) is a bulky network mainly riding on LSTM-based attention modules. The Global Feature Map Attention (GFMA) explores the inter-channel dependency at the early stage of ResNet to generate the visual attention coding for the proper scaling of feature responses. Feature Spatial Relationship Attention (FSRA) which runs at the later stage refines the feature representations by monitoring the state-to-state transition of feature maps via ConvLSTM. There is a need to prune ConvAM massively from 500M parameters to enable a wider range of applications.

Recurrent Attention Unit (RAU) (Ma & Boukerche, 2020) performs iterative feature map refinement while pushing the image through the convolution layers. Underlying it, three operations are executed. They are feature integration, attention mask generation, and attention state generation. The goal of RAU is to enhance the attention state matrices incessantly by distilling key information from small-scale and large-scale feature maps. Together with the fully connected layer of the backbone network, the attention state

matrices are used for logit computation and the whole network is optimized by cross entropy loss.

Boukerche and Ma (2021) elevate the efficiency of RAU by eliminating the redundant feature integration stage. The resultant module called Lightweight RAU (LRAU) is more lightweight and renders better accuracy. During the attention mask generation stage, LRAU consolidates the distinctive information through a joint evaluation of input attention state and scale-specific information. The generated attention mask then refines the attention state matrices to augment the feature perception ability. Similar to RAU, the attention state at the last time sequence and the top-level features of the backbone CNN are used to recognize the vehicles. The utilization of 1×1 , stride 2 convolution during attention state generation negatively impacts the learning capability of LRAU due to the loss of information. Better performance can be obtained if it is duly addressed.

Global Topology Constraint Network (GTCNet) (Xiang et al., 2019), a network built upon DenseNet264 (Huang et al., 2017), studies inter-part interactions for better VMMR performance. It comprises a point-wise convolution that assembles the deep abstract features learned from backbone CNN and a depth-wise convolution that reinforces the topology constraint by examining the global topology relationship between relevant parts. Since the topology relationship is computed for the top-level feature maps alone, the subtle features from the shallow layers are neglected and it leads to performance degradation (P. Wang et al., 2022).

A novel attention-based pooling dubbed Channel-wise Max Pooling (CMP) is introduced by Ma et al. (2019). To render rich embeddings, CMP pools a few largest activation values along the channel axis. Using DenseNet161 (Huang et al., 2017) as the backbone, it demonstrates encouraging performance

In Context-aware Attentional Pooling (CAP), Behera et al. (2021) amalgamated the contextual information from a few integral regions. Based on the predefined bounding boxes, 27 integral regions with different aspect ratios and sizes are sliced from the final feature maps and converted to fixed-size vectors via bilinear interpolation. The context-aware attention then recalibrates the feature vectors conditioned on the correlation between the integral regions and the neighbors. To extract the sequence-to-sequence information among the integral regions, the context vectors are sent in turn into LSTM for spatial structure encoding purposes. As a transition layer to the classification head, learnable pooling is utilized to summarize the information carried by the hidden state sequence of LSTM. The study also reveals that the classification performance can be highly swung by the number of integral regions. Future work should investigate viable ways to optimize it and thus avoid the need for human input.

Figure 2.18 depicts the ML-Decoder (Ridnik et al., 2021b), an attention-based classification head that is lightweight and highly versatile. Modified from the transformer decoder classification head (Vaswani et al., 2017), it successfully reduces the quadratic complexity by discarding the redundant self-attention operations. It relies on cross-attention and FFN in modeling the global dependencies. More importantly, it has high scalability against the high number of classes due to the existence of group fully connected layer.



Figure 2.18: ML-Decoder (Ridnik et al., 2021b)

Most of the studies in the attention domain go no further into inspecting the learned attention maps. In Counterfactual Attention Learning (CAL) (Rao et al., 2021a), the

efficacy of the learned attention maps is examined by comparing them against the random attention maps. The methodology is akin to Liu et al. (2020) who inspect the difference between the prominent and redundant regions of an object. To quantify the efficacy, the difference in logits produced from learned and random attention maps is computed and such insight is used as one of the supervisory signals to guide the learning process. As BP is employed as the pooling method, the classification head becomes extremely heavy, especially under a high number of classes.

As seen in Hu et al. (2019), Ji et al. (2023), Behera et al. (2021) and Rao et al. (2021a), BP is one of the pooling methods that is highly sought after due to its strong encapsulation capability. However, it is accompanied by an exponential increase in the number of parameters. In the Deep BP network (Du et al., 2023), a factorized BP is introduced as an alternative where low-rank matrices are used to approximate the bilinear operation and it results in massive parameter reduction. In addition, the Deep Network (DN) module is incorporated for implicit interaction among the feature responses through convolution, BN, non-linear layer and global pooling layer. For a more holistic prediction, the logits generated by both modules are taken into account. As both modules are highly distinct, the Deep BP network requires a complex training pipeline where the modules are trained individually before proceeding into joint training to ensure convergence of loss function.

To sum up, various forms of attention modules have successfully advanced vehicle recognition by performing feature calibration along the spatial axis, channel axis or by recursively refining the feature maps to underscore the discriminative information. Nonetheless, they still rely on convolution operation which examines the local instead of global relationship for attention computation and this incapacitates the feature reweighting process. Table 2.9 summarizes the attention mechanisms unraveled in this section.

Table 2.9: Summary of Attention-Based Methods

Reference	Method	Backbone	Dataset	#Classes	Metric	Highlight
Elkerdawy et al. (2018)	Co-occurrence Learning	ResNet50	CompCarsWeb	431	95.6%	Exploit the correlation between feature maps. The channel reduction layer causes information loss and leads to performance degradation.
Han et al. (2018)	A3M	ResNet50	CompCarsWeb	431	95.4%	Category-guided and attribute-guided attention modules use category and attribute information to jointly determine the prominent part features.
Zhuang et al. (2020)	APINet	DenseNet161	Stanford Cars	196	95.3%	Cross-image learning allows the discriminative features from pairwise counterparts to be utilized to complement the learning of key object parts. A proper pair construction strategy needs to be implemented for effective training.
Gao et al. (2020)	CIN	ResNet50	Stanford Cars	196	94.1%	Model cross-image channel interaction to mine unique features. A proper image pairing strategy is needed for training purposes.
Yu et al. (2020)	ConvAM	ResNet50	Stanford Cars	196	93.1%	GFMA examines inter-channel relationships via LSTM whilst FSRA enforces spatial relationships of local features via ConvLSTM. ConvAM has a huge number of trainable parameters.
			CompCarsWeb	431	95.3%	
Ma and Boukerche (2020)	RAU	ResNet101	Stanford Cars	196	93.8%	Refine feature maps iteratively to retain cross-granularity information. Less parameter-efficient as similar operations are seen in feature integration and attention mask generation stages.
			CompCarsWeb	431	97.8%	
			CompCarsSV	281	98.9%	
Boukerche and Ma (2021)	LRAU	ResNet50	Stanford Cars	196	93.9%	Generate and refine attention masks through recurrent architecture. 1×1 , stride 2 convolution during attention state generation causes information loss.
			CompCarsWeb	431	98.3%	

Table 2.9: Summary of Attention-Based Methods, continued

Reference	Method	Backbone	Dataset	#Classes	Metric	Highlight
Xiang et al. (2019)	GTCNet	DesneNet264	Stanford Cars	196	94.3%	Enforce spatial topology relationship between vehicle parts through depth-wise convolution. Only global feature maps are considered in computing spatial topology relationships.
			CompCarsWeb	431	98.5%	
Ma et al. (2019)	CMP	DenseNet161	Stanford Cars	196	93.7%	Reduce feature map depth by considering only salient features along the channel axis. Prone to overfitting.
			CompCarsWeb	431	97.9%	
Behera et al. (2021)	CAP	Xception	Stanford Cars	196	95.6%	Exploit inter-region relationships. Model performance is affected by the number of integral regions.
Ridnik et al. (2021b)	ML-Decoder	TResNet-L	Stanford Cars	196	96.4%*	ML-Decoder is a multi-purpose head for single-label classification, multi-label classification, and zero-shot learning. Its complexity is invariant to the number of classes and is useful for fine-grained classification.
Rao et al. (2021a)	CAL	ResNet101	Stanford Cars	196	95.5%	Examine the quality of learned attention and use it as a supervisory signal to better the learning. Large embedding size causes it to have low training efficiency especially when the dataset has a large number of classes.
Du et al. (2023)	Deep BP	ResNet34	Stanford Cars	196	91.8%	The bilinear module approximates BP through a low-rank matrix to reduce the complexity. A complex training methodology is required for optimum learning.

2.4.2.7 Multi-Scale Features

The microscopic details are the vital differentiators for the vehicles. Although these details are captured in the early layers, they are slowly diluted by the convolution operation and eventually overshadowed by the macroscopic elements from high-level feature maps (Tian et al., 2021). In this regard, the vehicle recognition performance can be less compelling if only the top-level information is harnessed and the low-level feature maps that embed the fine-grained details are left out. Therefore, considering both types of feature maps is a wiser act that leads to more conclusive predictions. This section expounds on the works that deliver fascinating performance upon leveraging multi-scale feature maps.

Progressive Multi-Granularity (PMG) (Du et al., 2020) employs multi-level classifiers at every pyramid level. The main intention is to allow the logits from the shallow and deep layers to collectively determine the vehicle class. Furthermore, a jigsaw puzzle strategy is introduced to enhance the learning of scale-specific information by shuffling the image patches. To compute scale-specific logits, there is a total of 4 forward passes required for an image and this translates to high Floating Point Operations (FLOPs) (Wu et al., 2022)

A cross-image semantic feature learning is studied in Cross-X learning (Luo et al., 2019). It comprises the One-Squeeze Multi-Excitation (OSME) that tunes the activation values based on the channel significance at individual scale levels. Along the process, a Cross-Category Cross-Semantic (C3S) regularizer is deployed to ensure similar semantic understandings among the feature maps from the same excitation module. To synthesize multi-scale feature maps, the Feature Pyramid Network (FPN) (Lin et al., 2017a) consolidates information from adjacent scales. A Cross-Layer (CL) regularizer that utilizes KL divergence loss then distills the knowledge between the mid-level, high-level,

and FPN features. Despite the perception that multi-scale features are more comprehensive and inclusive, the performance of Cross-X learning is slightly poorer than PMG which processes the scale-specific feature maps separately.

In the Attention Pyramid CNN (APCNN) (Ding et al., 2021), aside from FPN, a lateral connection connecting FPN and AP is established. AP synthesizes the multi-scale features further by calibrating the feature maps with spatial and channel attention in the bottom-up pathway. Moreover, the ROI-Guided Zoom-In and ROI-Guided Dropblock which perform vehicle localization and erase prominent regions, respectively are adopted for a better learning process.

Multi-Scale Discriminative Regions Attention Network (MS-DRAN) (Rong et al., 2021) also harnesses the multi-granularity features through FPN. With the help of backpropagated gradients with respect to the shallow-layer features, the discriminative region is identified and this narrows the focus area for the subsequent layer. The training process is supervised by focal loss and interclass ranking loss.

FPN is a highly popular technique in the domain of multi-scale features as evident in Ding et al. (2021), Luo et al. (2019) and Rong et al. (2021). Nevertheless, it is not impeccable as it only considers the information in immediate proximity to the current scale level. Besides, the problem of feature-scale confusion is not addressed and it leads to a tilted balance between the deep-layer features over the shallow-layer features.

Referring to Figure 2.19, Cross-Part CNN (CP-CNN) (M. Liu et al., 2021) addresses the scale dominance problem of FPN with Feature Enhancement Block (FEB). Instead of the simple addition and upsampling operation, the FEB injects the correlation information from the shallow layer into the deep layer to render a balanced proportion between the two. Thereafter, the multi-scale features serve as an input to the object parts proposal

module that leverages Grad-CAM for parts localization. The Context Transformer (CT) sequentially refines these parts via the Cross-Feature Augment module and the parts are eventually consolidated in the Pyramid Context Block (PyCB). CP-CNN requires 2 forward passes for execution of object localization and cross-part learning.

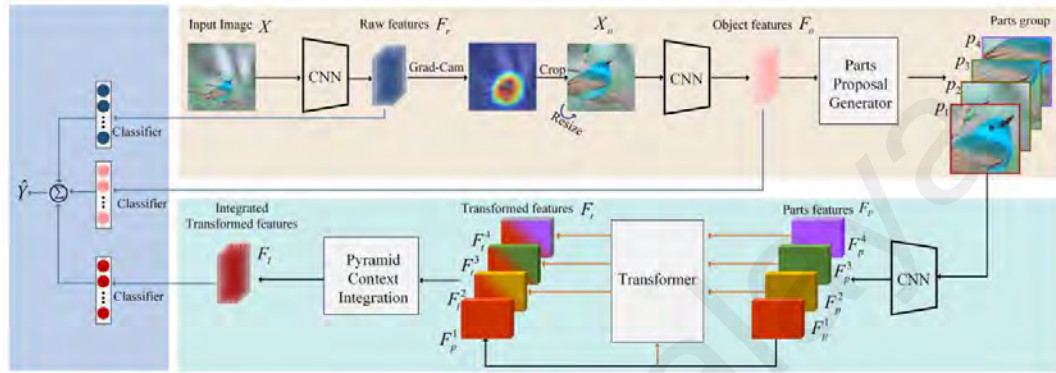


Figure 2.19: Cross-Part CNN (M. Liu et al., 2021)

Multi-scale feature maps can be generated by blending the cross-layer information (Ding et al., 2021; M. Liu et al., 2021; Luo et al., 2019; Rong et al., 2021). Another alternative, as taken by G. Wang et al. (2021) and L. Wang et al. (2022), executes convolutions with multiple receptive field sizes to achieve the same purpose.

Multi-Scale Pyramid Convolution (MSPyConv) (G. Wang et al., 2021) ameliorates the convolution operation of ResNet (He et al., 2016) by adding multi-kernel convolution. An attention module that covers both channel and spatial attention is also included to refine the feature maps based on saliency. The solution has high computational complexity due to the utilization of large-size convolution kernels.

Similarly, L. Wang et al. (2022) revised the last building block of ResNet34 (He et al., 2016) into Parallel Convolutional Block (PCB) which executes multi-size and multi-stream convolution. The features produced by each layer within the PCB are then fused through the matrix outer product and normalization operation in the Multilayers Feature

Fusion (MFF) layer. As PCB and MFF are limited to top-level feature maps, the learned feature embeddings are still deprived of low-level details.

Multi-Scale Sparse Network with Cross-Attention mechanism (CA-MSNet) (Maopeng Li et al., 2022) equips the bottleneck block of ResNet50 (He et al., 2016) with multi-scale features learning capability. The bottleneck block is revamped into the MSS module that exerts a multi-scale receptive field. For focused learning, the modular CA mechanism module is utilized by CA-MSNet to pay attention to the discriminative spatial information on the feature maps. The number of parameters for CA-MSNet (44.2M) is almost double that of the backbone ResNet50 (23.9M).

The Multi Squeeze-Excitation (MultiSE) (Yu et al., 2022) block in the ResNet50-based Vehicle Model Classification Subnetwork executes dilated convolution with multiple dilation rates to widen the field of view of the convolution kernels. In addition, it enriches the feature representation by exploiting the vehicle viewpoint embeddings extracted from the Tiny-YOLOv3 (Pose Estimation Subnetwork) (Adarsh et al., 2020). Being named collectively as Embedding Pose CNN (EPCNN), their performance is exceptional, particularly on the CompCarsWeb dataset but this comes at the expense of additional viewpoint labels aside from the class labels.

Unlike PMG (Du et al., 2020) which is disturbed by the noisy low-level classification head, Granularity-aware Distillation and Structure Modeling Region Proposal Network (GDSMPNet) (Ke et al., 2023) resorts to the intermediate and top-level feature maps for logits computations in the Granularity-Aware Distillation module. To align the semantic understanding between different hierarchical levels, Cross-Layer Self-Distillation regularization guides the early layers in terms of discriminative regions by scaling the activation values of low-level feature maps with the predicted class probability from high-level feature maps. The Structure Modeling Region Proposal module performs part

localization through feature maps binarization (Hu et al., 2019; F. Zhang et al., 2021; L. Zhang et al., 2021) and structure modeling loss is responsible for parts synthesis by optimizing the distance and angle between various parts in the polar coordinate system.

Feature Relocation Network (FReNet) (Zhao et al., 2023) features a Distractive Feature Learning (DFL) module to sideline the distractive features by employing distractive loss to achieve l_2 distance maximization between the high-level and low-level feature maps. In a separate module called Relocated High-Level Feature Learning (RHLFL), the mid-level features undergo similar operations to relocate the high-level features and learn the local details. The prediction is jointly determined by the refined mid-level and top-level feature maps. DFL and RHLFL should be used appropriately to prevent severe performance degradation since as high as 1.7% accuracy drop is seen in their experiments.

In a nutshell, a blend of macroscopic and microscopic components has delivered impressive vehicle recognition performance but a resolution to render an appropriate mix between the two is a pressing issue to solve. An ineffective multi-scale feature learning module causes the high-level semantic information to overshadow the low-level fine-grained details which in turn reduces the differentiation ability of the resultant network. Table 2.10 summarizes the literature utilizing multi-scale features.

Table 2.10: Summary of Multi-Scale Features-Based Methods

Reference	Method	Backbone	Dataset	#Classes	Metric	Highlight
Du et al. (2020)	PMG	ResNet50	Stanford Cars	196	95.1%	PMG features a jigsaw puzzle strategy to learn multi-scale features in stages. Multiple forward passes are needed.
Luo et al. (2019)	Cross-X Learning	ResNet50	Stanford Cars	196	94.6%	C ³ S aligns the embeddings from the same excitation modules semantically. CL aligns the logit from cross-layer features. FPN has feature-scale confusion problem.
Ding et al. (2021)	AP-CNN	ResNet50	Stanford Cars	196	95.3%	Use FPN and AP as dual path structures to extract and refine multi-scale features. ROI-Guided Dropblock and ROI-Guided Zoom-In prevent overfitting and perform localization, respectively. High-level features overshadow low-level features in FPN.
Rong et al. (2021)	MS-DRAN	ResNet50	Stanford Cars	196	94.3%	Multi-scale feature maps are refined using backpropagated gradient. FPN does not consider low-level and high-level features equally.
			CompCarsWeb	431	98.1%	
M. Liu et al. (2021)	CP-CNN	ResNet50	Stanford Cars	196	95.4%	FEB addresses scale confusion of FPN in integrating multi-scale features. CT allows part interaction based on significance. PyCB integrates part information efficiently. Require two forward passes.
G. Wang et al. (2021)	MSPyConv	ResNet50	Stanford Cars	196	93.6%	Employ various sizes of convolution kernels to deduce multi-scale features. Multi-attention modules are added for spatial and channel recalibration. Large convolution kernels are used.
L. Wang et al. (2022)	PCB & MFF	ResNet34	Stanford Cars	196	93.4%	PCB features multi-stream convolution operations and MFF consolidates cross-layer information. Loss of low-level details is still significant as multi-scale features only originate from the last convolutional block of ResNet.
Maopeng Li et al. (2022)	CA-MSNet	-	Stanford Cars	196	93.5%	MSS captures subtle features through multi-scale receptive fields. CA highlights the discriminative spatial positions. Large model size.
Yu et al. (2022)	EPCNN	Tiny-Yolov3, ResNet50	Stanford Cars	196	94.6%	MultiSE computes multi-scale features. Use viewpoint information to increase the classification performance. Require viewpoint annotation as the label.
			CompCarsWeb	431	98.9%*	

Table 2.10: Summary of Multi-Scale Features-Based Methods, continued

Reference	Method	Backbone	Dataset	#Classes	Metric	Highlight
Ke et al. (2023)	GDSMPNet	ResNet50	Stanford Cars	196	95.3%	Perform cross-layer KD. Integrate discriminative parts based on the polar coordinate system.
Zhao et al. (2023)	FReNet	ResNet101	Stanford Cars	196	95.4%	Improve the feature extraction process by contrasting high-level and mid-level features with low-level features. Overuse of DFL and RHLFL deteriorates classification performance by a large margin.

2.5 Summary

The development of AI has advanced vehicle recognition solutions. The sensor-based frameworks which are normally built upon machine learning algorithms deliver decent performance but the applications are limited to coarse-grained classification. More studies are required to investigate their feasibility for fine-grained classification tasks, which often go beyond 100 classes. More importantly, the sensors need to comply with operating conditions such as optimum temperature, low traffic volume and little to no vehicle occlusion. They also demand periodic maintenance and calibration for ideal performance.

Fortunately, the emergence of CV provides an alternative and is applicable for use cases that require the segregation of vehicles at granular levels i.e. VMMR. Within CV, the implementation of model-based solutions is intractable since the 3D vehicle modeling process is intricate. The feature-based methods, despite their encouraging results, are highly criticized for their dependency on handcrafted features which greatly reduce the solution robustness against environmental disturbances. Under the novel backbones category, two architectural variants, which are CNNs and transformers, are studied for general image classification tasks and they can be customized further to suit vehicle recognition tasks. The unsupervised filter learning techniques serve as a replacement for backpropagation for the optimization of convolution kernels. However, such a technique leads to an underperforming CNN because the optimization process does not take in the ground truth labels that drive the kernels toward maximizing the class separation in the feature space.

The part-based methods learn localization and classification tasks in a unified framework. Most of them are computationally expensive since the localization process requires additional forward passes. The attention category features several attention

modules as a drop-in mechanism to enhance the learning of vehicle recognition by focusing on discriminative regions. As they are built on convolution architecture, the attention span is limited to the close vicinity of the current spatial position. Therefore, there is a need for a fair feature recalibration process that evaluates feature importance at a global level. For the multi-scale features category, multiple learning schemes to consolidate low-level and high-level features are seen. Nevertheless, the extant multi-scale features synthesis process fails to render a balanced spread among features from different pyramid levels due to the repetitive passing on of features from early layers to the classification head. An innovative multi-scale feature learning paradigm is needed to bring out the full potential of coarse-to-fine-grained features.

CHAPTER 3: PCA-LDA-BASED CONVOLUTIONAL NEURAL NETWORK WITH CHANNEL-BASED ATTENTION MODULE FOR VEHICLE MAKE AND MODEL RECOGNITION

3.1 Introduction

Principal Component Analysis (PCA) is a highly sought-after unsupervised filter learning technique that serves as a workaround for the time-consuming backpropagation technique in optimizing convolution kernels of Convolutional Neural Networks (CNNs). As evident in Huang et al. (2015), Soon et al. (2018) and Soon et al. (2020), there is a formidable improvement in terms of accuracy and training duration by adopting PCA filters. However, the PCA filters are criticized for being class-agnostic and this limits the capacity to detect the minute interclass differences for Vehicle Make and Model Recognition (VMMR) purposes (Fang et al., 2016).

In this chapter, as a measure to generate class-sensitive convolution kernels, Linear Discriminant Analysis (LDA) which leverages the target variable as the supervisory signal is added to the filter learning pipeline. The LDA maximizes the class separation in the feature space and eventually produces convolution kernels that are more class discriminative. Besides, a customized attention module, namely the Channel-Based Attention Module (ChBAM), is incorporated for channel refinement purposes to increase the feature perception ability.

Overall, the proposed framework is designed to leverage CaffeNet (Jia et al., 2014) for deep feature extraction purposes. The extracted deep features undergo further processing by convolving with PCA and LDA filters. The utilization of PCA and LDA produces superior filters that pay attention to data variance and class separation and result in more discriminative features. Furthermore, the produced feature maps are reweighted by

ChBAM to increase the contribution of important channels in logit computation. The contributions of this chapter are summarized as follows:

- Propose an end-to-end PCA-LDA-CNN for VMMR that is accurate and fit for real-time implementation
- Inspect the impact of ChBAM which performs channel refinement toward classification performance
- Prove that the proposed framework is able to preserve the discriminative property of a vehicle and is robust against various distortions

3.2 Literature Review

VMMR is an arduous fine-grained classification problem that classifies a vehicle to the most granular level despite the huge vehicle design variations. Petrovic and Cootes (2004) performed classification for 77 vehicle models based on normalized structures of vehicle frontal image and achieved 93.0% accuracy but their method is not robust against rotation and lighting. Clady et al. (2008) performed VMMR for 50 vehicle models based on oriented contour points. They employed three voting algorithms and distance error for classification and reported 93.1% accuracy. Psyllos et al. (2009) proposed a framework with 87.0% accuracy for 11 vehicle models using Scale Invariant Feature Transform (SIFT) as the feature extractor and nearest neighbours (NN) as the classifier. Pearce and Pears (2011) applied Locally Normalized Harris Strength (LNHS) and Naïve Bayes classifier on 262 frontal images covering 74 vehicle models and reported 96.0% accuracy. Zhang (2012) adopted the Gabor Wavelet Transform (GWT) and Pyramid Histogram of Oriented Gradient (PHOG) as feature extractors. The method recorded 98.7% accuracy for 21 vehicle models using cascade classifier ensembles. The framework also includes a reject option to suppress the result when the confidence score is low. The ability of the framework for real-time application cannot be assessed as no inference speed is reported.

Siddiqui et al. (2016) proposed a Bag of Speeded-Up Robust Features (BoSURF) to classify 29 vehicle models. The reported accuracy was 94.8% but their method is affected by light illumination. Manzoor and Morgan (2017) quantized SIFT keypoints with Bag of Features (BoF) to enhance the explanatory power for VMMR purposes. Although having 89.1% accuracy, it is not feasible for real-time applications due to the long computational time of SIFT. Despite the exceptional performance of these works, the number of classes that they are tested against is less than 100, which is insignificant compared to the number of vehicle models out there. Moreover, most of them have a strong dependency on handcrafted features. Therefore, they are not robust against distortion such as lighting, rotation and viewpoint, which is commonly found in surveillance-nature images (Huang et al., 2015; S. Li et al., 2018).

CNN eradicates the feature handcrafting process owing to the convolution operation. The convolution operation studies the local region and progressively assimilates the local features to form highly robust global features. This allows the whole vehicle to be characterized holistically and results in better discrimination from one vehicle to another. Using the physical dimension of the vehicle as input, Zhu and Guo (2012) proposed a radial basis function (RBF) neural network. Although they achieved 96.7% accuracy, the process of gathering the physical measurements of the vehicles for model training purposes is daunting. Yang et al. (2015) performed VMMR using various vehicle viewpoints. It is concluded that the full-view carries the richest information with 76.7% accuracy. Dong et al. (2015) implemented a semi-supervised CNN with Sparse Laplacian Filter Learning (SLFL). They also introduced multi-task learning to train the softmax classifier and reported 88.1% accuracy. By using PCA filters, Huang et al. (2015) presented a CNN that delivered 99.1% accuracy for Vehicle Logo Recognition (VLR). Their method is known due to the enormous training time reduction as compared to backpropagation. A part-based VMMR approach was proposed by Fang et al. (2016).

They realized the automatic detection of useful vehicle parts by observing the activation values of the feature maps. They reported 96.6% accuracy by inferencing on multiple vehicle parts such as whole, center, left and right parts of the vehicle. Y. Li et al. (2018) proposed a Vehicle Type Recognition (VTR) framework based on the saliency map and ResNet (He et al., 2016). They delivered 94.1% accuracy for 3 vehicle types.

3.3 Methodology

3.3.1 Feature Extractor

Since the raw pixel information of a vehicle image can be noisy, CaffeNet (Jia et al., 2014) is exploited as an initial processing step to deduce discriminative and highly robust deep features before proceeding to PCA-LDA-CNN. The architecture of CaffeNet is shown in Figure 3.1.

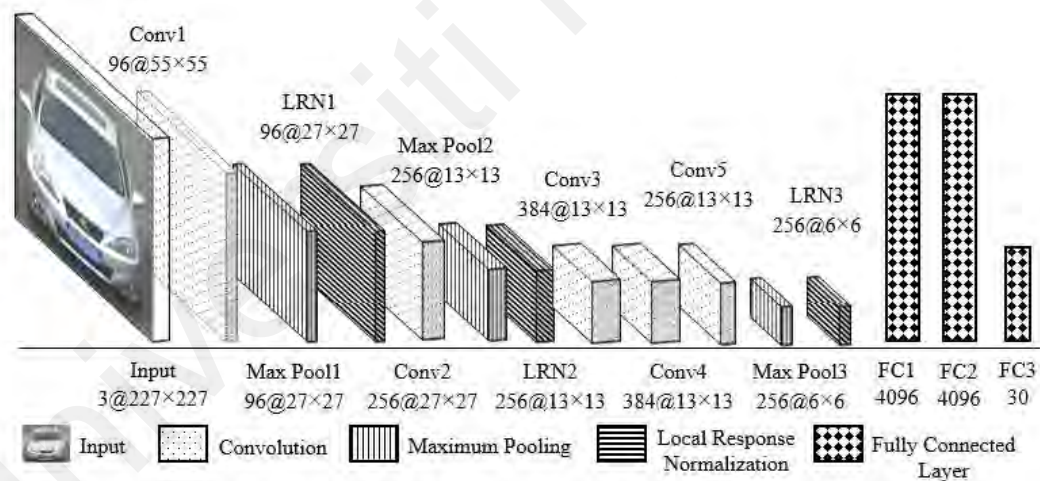


Figure 3.1: CaffeNet

There are 5 convolution layers, 3 Local Response Normalization (LRN) layers and 3 fully connected layers. The convolution kernels capture the local relationship within the local receptive field and subsequently aggregate the local features into global features. The convolution kernels are shared across the entire feature maps and the number of trainable parameters is reduced as a result. Following the convolution layer, rectified linear unit

(ReLU) injects non-linear signals and maximum pooling downsizes the spatial resolution of the feature maps. The LRN layer then normalizes the feature response over local regions for better generalization. Denoting a feature response of feature maps from i^{th} channel as a_i , the normalized feature response b_i is given by

$$a_{i,norm} = \frac{a_i}{\left(b + \alpha_{LRN} \sum_{j=\max(0,1-\frac{r}{2})}^{\min(C-1,i+\frac{r}{2})} a_j^2 \right)^{\beta_{LRN}}} \quad (3.1)$$

where C is the number of channels, bias $b = 2$, alpha $\alpha_{LRN} = 10^{-4}$, beta $\beta_{LRN} = 0.75$ and radius $r = 5, 2, 5$ for LRN1, LRN2 and LRN3, respectively. The last fully connected layer has L number of neurons where L is the number of classes.

When the training is completed, all training images are put through the network for feature extraction. For each training image, the tensor from the FC2 is extracted and reshaped into $64 \times 64 \times 1$ before being forwarded to the PCA-LDA-CNN for filter learning and inference.

3.3.2 PCA-LDA-CNN

The architecture of PCA-LDA-CNN is presented in Figure 3.2. It is a shallow CNN that comprises three convolution layers. It takes in the features extracted from CaffeNet and predicts the corresponding vehicle models. The proposed framework is partially inspired by the noise invariant and lightweight PCANet (Chan et al., 2015) which learns convolution kernels using PCA. The uniqueness of the proposed work is on top of PCA filters, filters are generated using LDA which aims to maximize class separation. A batch normalization (BN) layer is also added to avoid covariate shifts. Non-linearity and downsizing of feature maps are carried out with ReLU and maximum pooling. More importantly, a parameter-free ChBAM is inserted into PDA-LDA-CNN to reweight the

feature maps based on the information saliency. A fully connected layer and softmax classifier are used at the end to perform VMMR.

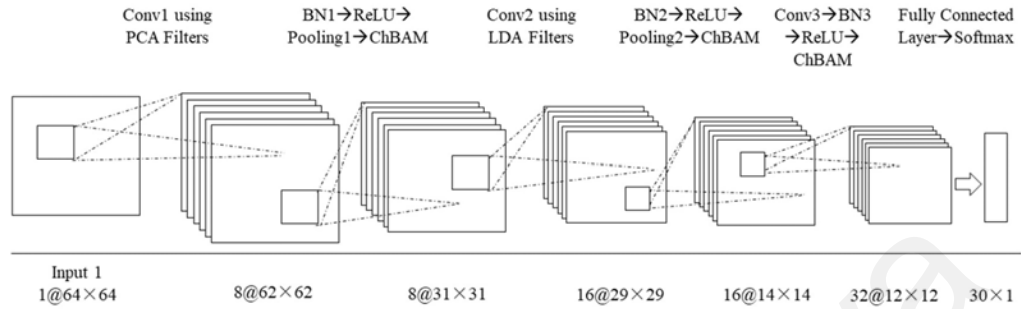


Figure 3.2: PCA-LDA-CNN

3.3.2.1 Generation of PCA Filters

Given input $\{I_i\}_{i=1}^{N_{Train}} \in \mathbb{R}^{H \times W \times C}$ where N_{Train} is the number of training images and H, W, C and are height, width and number of channels, it is converted into a sequence of $k \times k \times C$ patches and the feature responses are subtracted by the mean of the patch. The patch sequence of I_i is represented as

$$I_i^{patch} = \{I_{i,1}^{patch}, I_{i,2}^{patch}, \dots, I_{i,P}^{patch}\} \in \mathbb{R}^{P \times (k \times k \times C)} \quad (3.2)$$

where $P = [(H - k)/s + 1]^2$ and convolutional stride s is 1. For brevity, I_i^{patch} is denoted as X_i from here onwards. Cascading all X_i will then produce $X \in \mathbb{R}^{(N_{Train} \times P) \times (k \times k \times C)}$.

PCA filters are essentially the eigenvectors of $X^T X$. Suppose N_{PCA} is the number of PCA filters to be generated, the eigenvectors can be deduced by minimizing PCA reconstruction error as formulated by

$$\min_{G \in \mathbb{R}^{(k \times k) \times N_{PCA}}} \|X - XGG^T\|^2 \text{ s.t. } GG^T = I \quad (3.3)$$

where I is an identity matrix and $G = [G_1, G_2, \dots, G_{N_{PCA}}]$ is the eigenvectors. For PCA filters, $k = 3$ and $N_{PCA} = 8$ are chosen. At the Conv1 layer, the generated PCA filters are used to convolve with I_i , which is the vehicle features extracted from CaffeNet.

3.3.2.2 Generation of LDA Filters

I_i after passing through Conv1, BN, ReLU, maximum pooling and ChBAM has $31 \times 31 \times 8$ resolution. To generate LDA filters, the feature maps are also converted to X using Equation (3.2). Denoting S_l as the indices for the input under class l , the class mean μ_l and within-class scatter matrix $S_{w,l}$ are computed as follows:

$$\mu_l = \frac{1}{|S_l|} \sum_{i \in S_l} X_i \quad (3.4)$$

$$S_{w,l} = \frac{1}{|S_l|} \sum_{i \in S_l} (X_i - \mu_l)(X_i - \mu_l)^T \quad (3.5)$$

Subsequently, the overall mean μ and between-class scatter matrix S_b are calculated as

$$\mu = \frac{1}{L} \sum_{l=1}^L \mu_l \quad (3.6)$$

$$S_b = \frac{1}{L} \sum_{l=1}^L (\mu_l - \mu)(\mu_l - \mu)^T \quad (3.7)$$

Different from PCA which aims to maximize the variance, LDA maximizes class separability, which is the ratio of S_b to S_w . Suppose N_{LDA} is the number of LDA filters to be generated, the LDA filters can be deduced as follows:

$$\max_{E \in \mathbb{R}^{(k \times k) \times N_{LDA}}} \frac{\text{Tr}(E^T S_b E)}{\text{Tr}(E^T (\sum_{l=1}^L S_{w,l}) E)} \quad \text{s. t. } EE^T = I \quad (3.8)$$

where $Tr(\bullet)$ is trace operator and E is the eigenvectors of $(\sum_{l=1}^L S_{w,l})^{-1} S_b$. For the generation of LDA filters, k and N_{LDA} are set as 3 and 16, respectively.

3.3.2.3 Channel-Based Attention Module

ChBAM is inspired by the Squeeze-and-Excitation (SE) block (Hu et al., 2018). It performs transformation by reweighting feature maps across the channels. Essentially, a single output feature map is obtained by doing a weighted summation among the input feature maps from multiple channels. Therefore, the feature map is channel-dependent implicitly while at the same time spatially connected with neighboring pixels due to convolution kernels.

Figure 3.3 illustrates the proposed ChBAM. To exploit the channel dependency, global maximum pooling is first used to perform channel-based summarization and the maximum response from each channel z_c is obtained. Since the cross-channel information exchange has been covered by convolution, the fully connected layer following z_c , as evident in Hu et al. (2018), is discarded for the sake of simplicity. Hence, upon getting z_c , the sigmoid function normalizes the values between 0 to 1 and they are subsequently used to scale the feature responses. The benefit of ChBAM over SE block is it is completely parameter-free and no additional computational burden is required.

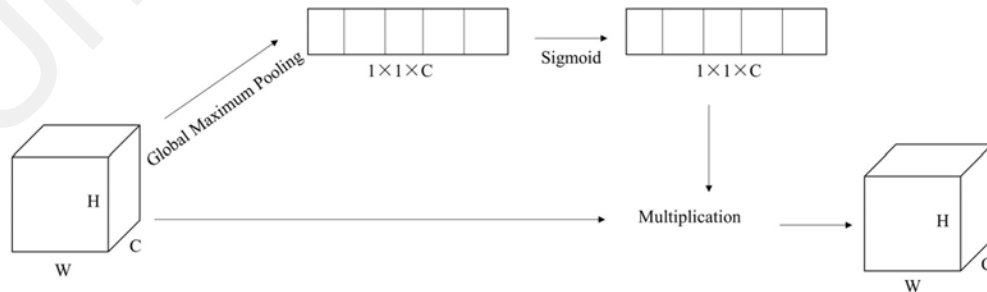


Figure 3.3: Channel-Based Attention Module

3.4 Experiments

3.4.1 Datasets

To validate the performance of the proposed methodology, a mixture of BVMMRv2 (Biglari et al., 2017b) and Surveillance-Nature Comprehensive Cars (CompCarsSV) (Yang et al., 2015) datasets are used. The images are captured by the surveillance cameras and they account for 4,858 and 44,481, respectively. As the images are affected by external disturbances such as different lighting conditions and noise, they serve as a good indicator of the performance of the framework under actual scenarios. As shown in Figure 3.4, the captured images from both datasets contain the full vehicle view. It covers various useful features, including the vehicle logo, headlight, and license plate.



Figure 3.4: Samples Images from BVMMRv2 and CompCarsSV

For BVMMRv2, since some images contain more than one vehicle, segmentation is carried out so that one image contains only a vehicle. The images categorized under ‘Other Classes’ are also dropped as it is beyond the interest of the study to predict the vehicles from unknown categories. Furthermore, upon analyzing the class distribution, it is realized that the quantity of ‘Pride 131’ is far larger than the rest. Hence, only half of the images are sampled and used for training purposes. To reduce the effect of imbalanced classes, the image count for ‘Lifan 620’ is increased by adding those from CompCarsSV. As for the classes with insufficient samples i.e. less than 25 images, they are replaced with randomly selected classes from CompCarsSV. An additional 3 classes are also picked from CompCarsSV to make the dataset richer. Random affine transformation is

employed as the data augmentation technique to increase the size of the dataset further and this makes the primary dataset have training images of 9,994 and testing images of 1,242, respectively, covering 30 makes and models with the 90:10 train-test ratio.

Since a reliable VMMR system should operate during both daytime and night-time, extra caution is taken by ensuring these images are available during the training and testing phases. Table 3.1 illustrates the proportion of night-time images in the primary dataset, which accounts for at least 7% of the training and testing images.

Table 3.1: Breakdown of Daytime and Night-time Images

Stage	#Daytime Images	#Night-time Images	Ratio of Night-time Images
Training	9,161	833	8.3%
Testing	1,146	96	7.7%

As the primary dataset is a partial mixture of BVMMRv2 and CompCarsSV, the framework is evaluated further by fully combining both datasets. With the 70:30 train-test ratio, the secondary dataset has 34,911 training and 14,268 testing images, covering 300 vehicle makes and models.

3.4.2 Implementation Details

Upon initializing CaffeNet with ImageNet 2012 weights, it is fine-tuned for 50 epochs on 227×227 resolution. Stochastic Gradient Descent (SGD) is chosen to optimize the network parameters. A learning rate of 0.01 is adopted for the first 25 epochs and it is dropped to $1e-4$ for the next 25 epochs.

For PCA-LDA-CNN, backpropagation and Adaptive Moment Estimation (Adam) (Kingma & Ba, 2014) are used to optimize the learnable parameters, including the convolution kernels weights of Conv3, BN and fully connected layer. Adam optimizer is set to have a learning rate of 0.001, the exponential decay rate for the first moment

estimate β_1 of 0.9, the exponential decay rate for the second moment estimate β_2 of 0.999 and epsilon ϵ of $1e^{-7}$. The training lasts for 100 epochs.

All the experiments are carried out on a machine with the specification of Intel Core i7-9750H 2.6GHz, 32GB RAM and NVIDIA Quadro T1000 4GB video memory.

3.5 Results & Discussions

3.5.1 Quantitative Analysis

Table 3.2 presents a breakdown in terms of training duration for the training pipeline. The training pipeline involves fine-tuning CaffeNet, generation of PCA and LDA filters as well as training of PCA-LDA-CNN. The total training time is 33 minutes using GPU. The fine-tuning of CaffeNet takes the most amount of time as it is the largest network among all, accounting for 81.8% of the total training duration.

Table 3.2: Training Duration of Proposed Framework

Stage	Epoch	Duration (Minutes)
Fine-tuning CaffeNet	50	27
Generation of PCA Filters	-	1
Generation of LDA Filters	-	4
Training of PCA-LDA-CNN	100	1
Total Duration		33

The proposed framework is benchmarked against other works in Table 3.3 using the following metric

$$Accuracy = \frac{TP}{N_{Test}} \times 100 \quad (3.9)$$

where TP is the total true prediction count and N_{Test} is testing image count. Since these works are not evaluated on the same datasets, the number of classes is attached in the table for better gauging of solution competency and the dataset complexity. Based on the primary dataset described in Chapter 3.3.1, the proposed framework reports 99.6%

accuracy. The works by Petrovic and Cootes (2004) and Siddiqui et al. (2016) are not robust against distortions. A high image quality is the prerequisite for their solutions to deliver good performance. Although Clady et al. (2008), Pearce and Pears (2011), Zhang (2012), Hsieh et al. (2014) and H. He et al. (2015) achieve satisfactory accuracy, a decline in performance is seen for some of them when tested on the secondary dataset which is larger in scale. The accuracies obtained by Psyllos et al. (2009) and Manzoor and Morgan (2017) are not high enough and more improvements should be made. For Fang et al. (2016), their solution is validated with CompCarsSV which covers 281 classes but the inference time is not reported. The performance of Manzoor et al. (2019) is promising and it comes with high computational speed. The solution by Jamil et al. (2020) is highly reliant on handcrafted features and is vulnerable to environmental disturbances. More extensive validation should also be done using a larger dataset.

Table 3.3: Performance Benchmarking on Primary Dataset

Reference	#Classes	Accuracy	FPS
Normalized Vehicle Structures+NN (Petrovic & Cootes, 2004)	77	93.0%	-
Oriented Contour Point+NN (Clady et al., 2008)	50	93.1%	-
SIFT+NN (Psyllos et al., 2009)	11	87.0%	-
LNHS+Naïve Bayes Classifier (Pearce & Pears, 2011)	74	96.0%	-
GWT, PHOG+Cascade Classifier Ensembles (Zhang, 2012)	21	98.7%	-
Symmetrical SURF+Ensemble Classifier (Hsieh et al., 2014)	29	98.0%	21.0
Multiscale Retinex+Neural Network (H. He et al., 2015)	30	92.5%	-
Coarse-to-Fine CNN (Fang et al., 2016)	281	98.6%	-
BoSURF+SVM (Siddiqui et al., 2016)	29	94.8%	7.0 -7.5
BoSIFT+SVM (Manzoor & Morgan, 2017)	37	89.0%	-
HOG+SVM (Manzoor et al., 2019)	35	97.9%	13.9
BRISK+HOG+Bag of Expression+SVM (Jamil et al., 2020)	29	98.4%	6.7
CaffeNet+PCA-LDA-CNN	30	99.6%	6.7

Table 3.4 shows a detailed breakdown of the prediction performance. The TP count stands at 1,237. There are a total of 5 false predictions FP and they come from Pride 132, Saipa Tiba, IKCO Samand Soren and Lexus LX. The accuracy of these 5 classes can be improved further when more training images are made available.

Table 3.4: Performance Breakdown Analysis on Primary Dataset

Make	Pride			Peykan	Zamyad	Peugeot			
Model	131	132	141	80	Truck	405	405 SLX	206	Pars
TP	200	46	32	97	25	157	24	49	65
FP	0	2	0	0	0	0	0	0	0
Acc	100	95.8	100	100	100	100	100	100	100
Make	Buck	Zhonghua		Saipa	Citroen	IKCO			Kia
Model	Regal	H330	H530	Tiba	Xantia	Samand	Samand Soren	Runa	Rio
TP	49	32	24	33	8	95	10	16	14
FP	0	0	0	1	0	0	1	0	0
Acc	100	100	100	97.1	100	99.0	100	100	100
Make	Lexus		Renault	MVM		Lifan		Mazda	Audi
Model	IS	LX	L90	530	315	620	X60	Truck	Q3
TP	20	30	37	7	5	34	21	8	19
FP	0	1	0	0	0	0	0	0	0
Acc	100	96.8	100	100	100	100	100	100	100
Make	Benz			$Accuracy = \frac{1,237}{1,242} \times 100$ $= 99.6\%$					
Model	R Class	A Class	S Class						
TP	35	19	26						
FP	0	0	0						
Acc	100	100	100						

The proposed framework undergoes further validation on the secondary dataset whereas the rest are evaluated on CompCarsSV only, which is a subset of the secondary dataset, as reported in their original works. Table 3.5 suggests that the proposed framework outperforms the frameworks proposed by Zhang (2012), Hsieh et al. (2014) and Biglari et al. (2017a). Comparing Tables 3.3 and 3.5, it is realized that the performances of Zhang (2012) and Hsieh et al. (2014) degrade significantly when being evaluated on a larger dataset. Since the proposed framework is comparable to that of Fang et al. (2016) in terms of accuracy, a comparison in terms of inference speed is made. The work of Fang et al. (2016) is reimplemented and it records an inference speed of 2 frames per second (FPS). The proposed framework reaches 6.7 FPS, which is at least 3 times faster. As Siddiqui et al. (2016) and Manzoor et al. (2019) suggest that a practical VMMR

system should process 5-6 FPS, the proposed framework gains an advantage for real-time application.

Table 3.5: Performance Benchmarking on Secondary Dataset

Reference	#Classes	Accuracy	FPS
GWT, PHOG+Cascade Classifier Ensembles (Zhang, 2012)	281	83.8%	-
Symmetrical SURF+Ensemble Classifier (Hsieh et al., 2014)	281	51.7%	21.0
Coarse-to-Fine CNN (Fang et al., 2016)	281	98.6%	2.0
Latent SVM+Cascade Classifier (Biglari et al., 2017a)	281	97.4%	0.01
CaffeNet+PCA-LDA-CNN	300	97.8%	6.7

3.5.2 Ablation Study

To justify the design of PCA-LDA-CNN, an ablation study that uses 100×100 grayscale vehicle images as input is carried out. Classification accuracy between 3-layer CNN and the networks that use PCA or LDA filters are compared. The architecture in Figure 3.2 is adopted for these networks with the exception that the ways of optimizing the convolution kernels are different. These networks are trained for 50 epochs and the highest classification accuracy based on test data is recorded. For the 3-layer CNN, all convolution filters are learned through backpropagation. For networks that use PCA filters or LDA filters, only the convolution filters in the Conv3 layer are learned via backpropagation. The accuracies of these networks are tabulated in Table 3.6.

Table 3.6: Ablation Study

Network	Accuracy
3-layer CNN Without ChBAM	98.6%
LDA-LDA-CNN Without ChBAM	97.5%
LDA-PCA-CNN Without ChBAM	98.8%
PCA-PCA-CNN Without ChBAM	98.8%
PCA-LDA-CNN Without ChBAM	99.0%
PCA-LDA-CNN With ChBAM	99.2%

It is observed that determining convolution filters through supervised and unsupervised manners proves to be a better option as compared to the backpropagation

technique except for the network that uses LDA only. Out of all strategies, learning convolution filters by PCA followed by LDA yields the best outcome, which is 99.0% accuracy. The efficacy of ChBAM is also validated where inserting ChBAM into PCA-LDA-CNN further elevates the classification accuracy by 0.2%, thus hitting 99.2%.

3.5.3 Robustness Test

A VMMR algorithm that is feasible for real-life scenarios must be robust against external disturbances. It is ineluctable that the vehicle images are either affected by lighting, exist in a slanted position, or appear at different distances from the surveillance cameras. For a proper examination of the robustness, the occurrence of such disturbances is simulated by introducing translation, rotation and scaling as well as adjusting the brightness and contrast of the images. Various degrees of distortions are applied to half of the testing images to investigate the robustness of the framework.

3.5.3.1 Translation

Half of the randomly sampled testing images are translated either horizontally, vertically or in both directions to simulate the condition in which the vehicles are not positioned at the center of the camera lens as shown in Figure 3.5.



Figure 3.5: Images Translated with (a) -10, -10, (b) -10, 10, (c) 10, -10, (d) 10, 10

The degree of translation varies from -10 pixels to 10 pixels. From Table 3.7, it is observed the model is robust against translation as all accuracies stay above 97%. It is also noticed that translating the images downward causes a more significant drop in

accuracy as compared to translating the images upward. This is due to the distinctive vehicle front, which carries the most crucial information, resides at the lower part of images. Hence, missing vehicle frontal information is more critical than missing vehicle roof information, which is at the upper part of the images.

Table 3.7: Robustness Test against Translation

		Horizontal Shift (Pixel)				
		-10	-5	0	5	10
Vertical Shift (Pixel)	-10	98.4%	98.7%	99.0%	98.2%	97.8%
	-5	99.0%	98.9%	99.0%	99.2%	99.0%
	0	99.2%	99.4%	99.6%	99.3%	98.6%
	5	98.2%	99.3%	99.0%	99.0%	98.2%
	10	97.7%	97.8%	98.1%	98.1%	98.1%

3.5.3.2 Scaling

To simulate the vehicle-to-camera distances, scaling factors ranging from 0.6 to 1.4 is applied. This is equivalent to diminishing and magnifying the vehicle by half of the original size. Examples of vehicle images upon scaling are shown in Figure 3.6.



Figure 3.6: Images with Scaling Factor of (a) 0.6 (b) 0.8 (c) 1.2 (d) 1.4

Based on Figure 3.7, the performance of the proposed methodology suffers when the scaling factor changes from 0.7 to 0.6 where it drops from 92.1% to 78.9%. Moreover, by comparing the drop in classification accuracy, zooming out the images has a more severe effect than zooming in due to the loss of minute vehicle traits. Overall, the proposed methodology is robust against scaling at least from 0.7 to 1.4 scaling factor.

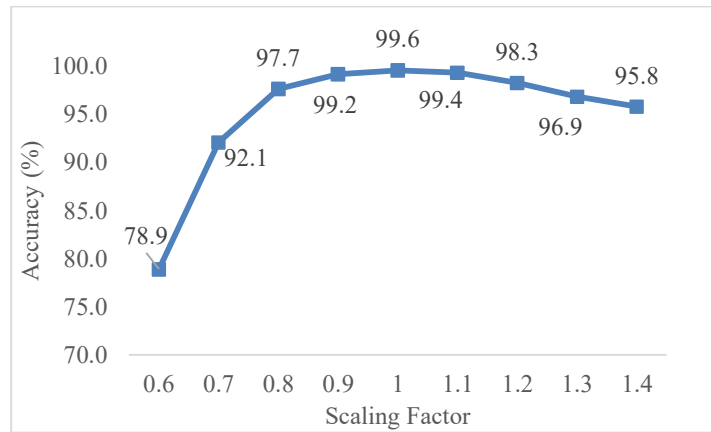


Figure 3.7: Robustness Test Against Scaling

3.5.3.3 Rotation

Half of the testing images are randomly rotated up to 20° either in the clockwise or counterclockwise direction. This is to put the proposed methodology under the test when the vehicles are coming from different angles with respect to the camera. Sample images upon rotation and the results of the rotation test are shown in Figure 3.8 and Figure 3.9, respectively. The proposed methodology is quite indifferent towards rotation. Given the images being rotated in the most extreme case, the classification accuracy still stays above 95%.



Figure 3.8: Images with Rotation (a) -20° (b) -10° (c) 10° (d) 20°

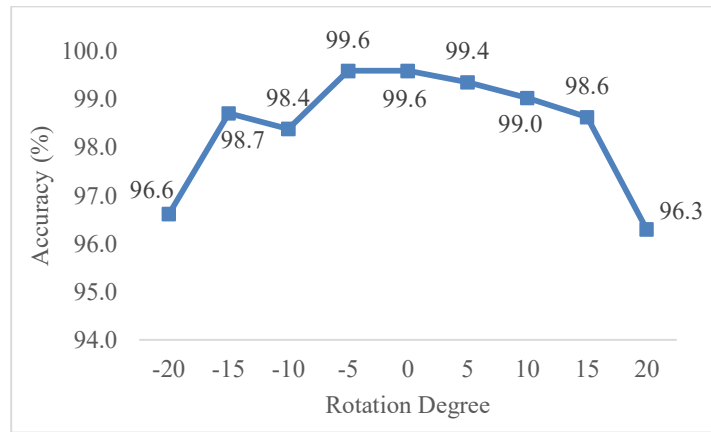


Figure 3.9: Robustness Test Against Rotation

3.5.3.4 Brightness and Contrast

As the weather is ever-changing and the vehicle images can be taken on sunny days, cloudy days or even rainy days, the light illumination is beyond control and this affects the quality of images. A robust algorithm should take this into account, and such conditions are simulated by adjusting the brightness and contrast of half of the testing images using the following equation

$$I'(x, y) = \lambda I(x, y) + \eta \quad (3.10)$$

where $I(x, y)$ represents the pixel value at (x, y) spatial position, $I'(x, y)$ is the resultant pixel value after adjustment and λ and η are the parameters to adjust contrast and brightness, respectively. The images after adjusting brightness and contrast are shown in Figure 3.10.



Figure 3.10: Images adjusted with λ and η of (a) 0.5, -20 (b) 0.5, 20 (c) 2, -20 (d) 2, 20

Based on the results in Table 3.8, the proposed methodology is more affected by the change in contrast than the change in brightness. Nevertheless, it is still robust against various degrees of brightness and contrast alteration as most of the classification accuracy stays above 90%. This result is within expectation as special precaution is taken to ensure a reasonable mix between daytime and night-time images within the training image pool.

Table 3.8: Robustness Test against Lighting and Contrast

Accuracy		λ			
		0.5	1	1.5	2
η	-20	95.9%	98.9%	99.3%	96.6%
	-10	97.8%	99.8%	99.2%	95.3%
	0	98.0%	99.6%	98.9%	93.6%
	10	97.8%	99.4%	98.2%	91.1%
	20	98.6%	99.4%	97.8%	88.6%

3.6 Conclusion

In this chapter, VMMR is addressed by proposing a framework that uses fine-tuned CaffeNet to extract distinctive features from full-view vehicle images. The images are then fed into PCA-LDA-CNN equipped with ChBAM for classification. The framework is benchmarked against the existing works using BVMMRv2 and CompCarsSV datasets and it delivers astounding accuracy with short inference time. The ablation study reveals that a CNN that adopts PCA and LDA filters for convolution acquires the highest accuracy as compared to backpropagation-optimized CNN, PCA-PCA-CNN, LDA-LDA-CNN and LDA-PCA-CNN. The effectiveness of ChBAM which carries out channel-based reweighting is evaluated and it brings marginal improvement even with zero trainable parameters. The framework also demonstrates excellent results in the robustness test and it is resilient enough to be deployed into actual scenarios.

CHAPTER 4: SPATIALLY RECALIBRATED CONVOLUTIONAL NEURAL NETWORK FOR VEHICLE TYPE RECOGNITION

4.1 Introduction

In this chapter, a detailed study of the attention domain is conducted to resolve Vehicle Type Recognition (VTR). VTR has been studied via various approaches, including feature-based, unsupervised filter learning and part-based techniques. Generally, these approaches ingest the top-level feature maps of Convolutional Neural Networks (CNNs) which are enclosed with semantically strong features for direct logit computation. Despite their remarkable performances, most of them do not implement differential learning which treats the feature responses according to information saliency. It is reckoned that every vehicle part does not assume the same importance level. They should be treated uniquely so that the discriminative parts are given higher attention whereas the inconsequential information is suppressed.

Aligned with this vision, a Spatial Attention Module (SAM) is proposed as an attention module to enhance the high-level features deduced from the convolution operation based on spatial importance. In SAM, the correlation of all spatial positions is quantified through an attention matrix that tracks global dependencies. The deduced attention maps are then used to remodel the top-level feature maps through element-wise multiplication and a softmax classifier is used to perform classification. The contributions of this chapter are stated as follows:

- Exhibit the ability of SAM to compute spatial relationships among global features and thus underscore the exclusive features
- Demonstrate SAM can be integrated into existing classification models to improve classification accuracy and it is trainable end-to-end

- Prove that SAM is better than existing attention mechanisms in terms of classification accuracy

4.2 Literature Review

The early Computer Vision (CV)-based VTR solutions exploited the raw vehicle features by examining the gradient orientation information of the vehicle. A cascade two-stage classifier ensemble was proposed by Zhang (2012) where Gabor Wavelet Transform (GWT) and Pyramid Histogram of Oriented Gradient (PHOG) are utilized to characterize vehicles. The feature vectors are subsequently fed into an ensemble of 25 models to perform classification through majority voting. Despite achieving 98.7% accuracy for 21 vehicle models, having numerous models may impact the feasibility of real-time implementation. Peng et al. (2012) applied a clustering technique for VTR. K-Means clustering is performed on the features deduced from Principal Component Analysis (PCA) and 88.8% accuracy was reported. Sun et al. (2017) derived global and local features from an improved canny edge detector and Gabor wavelet. A two-stage classification framework, namely the k-Nearest Neighbours Probability Classifier (kNNPC) and Discriminative Sparse Representation-Based Classifier (DSRC), is then used as the classifier. Their framework is tested on a limited number of images and an average accuracy of 93.0% was reported. Derrouz *et al.*'s work (Derrouz et al., 2019) is based on stereo vision. Using the disparity map generated from stereo vehicle images, the actual vehicle dimensions are derived. Next, HOG is applied to enrich the feature representation and the feature vector is downsized through PCA. Eventually, the feature vector together with vehicle dimensions serve as input to SVM. They reported 95.2% on the Beijing Institute of Technology (BIT)-Vehicle dataset (Dong et al., 2015). Sathyanarayana and Narasimhamurthy (2022) described vehicles through Gabor filters, HOG and Local Optimal Oriented pattern (Chakraborti et al., 2018). Then, Ant Colony Optimization (Parsons, 2005) is utilized to select the top 30% best features, thus reducing

features from 12,260 to 3,676 before feeding into a deep neural network. Their framework recorded 97.9% accuracy on the MIO-TCD dataset (Luo et al., 2018) and outperforms other deep CNNs such as ResNet50 (He et al., 2016) (96.9%), DenseNet (Huang et al., 2017) (97.0%) and Xception (Chollet, 2017) (97.6%). Y. Wang et al. (2019) improvised the Spatiotemporal Sample Consistency (STSC) algorithm to reduce lighting interference in the background subtraction technique during vehicle detection. The segmented vehicle is then fed into a cascade classifier to predict vehicle type based on the ratio of the license plate and vehicle dimensions, HOG features, passenger face as well as vehicle area. However, their network is not easily scalable as the inputs include the actual dimensions of license plates that can vary from country to country. Although these works report high accuracy, depending on handcrafted features causes them to be less robust. The network performance can be highly swung by translation, rotation, scaling and change in light illumination (Huang et al., 2015; S. Li et al., 2018).

Deep learning algorithms, particularly CNNs, underwent rapid development a decade ago due to their astounding learning capability. They perform feature extraction hierarchically and eventually generate the global features that carry strong semantic information. The advent of deep learning algorithms has lifted the experience-based feature engineering process and consistently achieved state-of-the-art performance. Jung et al. (2017) studied an ensemble technique based on deep learning models. They proposed Joint Fine-tuning (JF) to train several CNNs through joint loss function. They also implemented DropCNN to randomly drop CNN from the logits averaging process to prevent overfitting. An ensemble of 8 ResNet18 reported 98.0% accuracy on the MIO-TCD dataset. Boonsirisumpun and Surinta (2022) fine-tuned MobileNet (Howard et al., 2017) to differentiate 5 vehicle types and reported 93.4% accuracy.

Instead of optimizing convolution kernels through backpropagation, some studies suggest unsupervised filter learning techniques. Dong et al. (2015) proposed a semi-supervised CNN that learns convolution kernels through Sparse Laplacian Filter Learning (SLFL) and multitask learning. Their technique delivered 88.1% accuracy for 6 vehicle classes but is not discriminative enough between Sport Utility Vehicle (SUV) and sedan. Similarly, Huang et al. (2015) made use of PCA to deduce convolution kernels and the feature maps are used by SVM for classification. Their framework which delivered 99.1% accuracy on 10 vehicle makes has a longer inference time than CNN which uses backpropagation. The network proposed by Soon et al. (2020) also adopts PCA filters to derive vehicle features and 88.5% accuracy was attained for 6 vehicle types. In addition, Local Tiled CNN was proposed by Gao and Lee (2016) in which Topographic Independent Component Analysis is used to deduce the convolution kernels and 98.5% accuracy was reported. Although the unsupervised filter learning techniques show promising results, they are often disputed due to low robustness (Chan et al., 2015; Fang et al., 2016).

Some of the past studies employ part-based methods to address VTR. The two learning paradigms of part-based methods are supervised and weakly supervised techniques. After conducting training using bounding box annotations, Arinaldi et al. (2018) applied Faster Region-Based CNN which uses region proposal network, Region of Interest (ROI) pooling and convolutional architecture to carry out detection and classification for 6 vehicle classes. They reported 69.4% accuracy on the MIT Traffic dataset. With inference time as the primary focus, Tajar et al. (2021) performed pruning for YOLOv3-tiny to reduce the number of parameters. Zhao et al. (2022) improvised YOLOv4 (Bochkovskiy et al., 2020) by integrating the Convolutional Block Attention Module (CBAM) (Woo et al., 2018) and modifying Path Aggregation Network (S. Liu et al., 2018). Tajar et al. (2021) and Zhao et al. (2022) attained mAP 95.1% and mAP 83.5% for 6 vehicle types,

respectively. Rachmadi et al. (2018) modeled image classification as a sequence problem by attending to different parts of vehicles sequentially. After performing vectorization via ResNet18, every image partition as well as the original image are attended in turn by Long-Short Term Memory (LSTM). The solution achieved 98.0% accuracy on the MIO-TCD dataset. Another technique by Y. Li et al. (2018) is the combination of the compressed sensing technique and ResNet (He et al., 2016). Compressed sensing which has the advantage in terms of faster computational speed is used to generate a saliency map for vehicle detection and ResNet50 is used to carry out classification. Accuracies of 94.1% and 95.0% were reported for 3 vehicle classes based on MIT CBCL and Caltech Database, respectively.

Among the part studies from the attention domain, Ma and Boukerche (2020) proposed a Lightweight Recurrent Attention Unit (LRAU) that successively refines the feature maps based on the attention state matrices deduced from image pyramids. Despite reporting 93.9% accuracy on Stanford Cars (Krause et al., 2013), the utilization of 1×1 , stride 2 convolution to deduce attention states causes information loss. In SAM, the completeness of the learned contextual information is preserved by sending the top-level feature maps in full form into SAM to render more accurate classification results. In the Attention Pyramid CNN (APCNN) (Ding et al., 2021), the Feature Pyramid Network (FPN) (Lin et al., 2017a) is used to generate multi-scale features and they are further refined by the spatial and channel gates. APCNN achieved 95.3% accuracy on Stanford Cars but the learned features are suboptimal due to the limited receptive field of convolution kernels. With Multi-Head Self Attention (MHSA), SAM is able to track long-range dependencies and generate more holistic features. Attentive Pairwise Interaction Network (APINet) (Zhuang et al., 2020) identifies the salient region of the image by comparing and contrasting an object pair. A careful design of the image pair construction strategy is essential to ensure the convergence of the loss function. On the contrary, the

training pipeline of SAM is relatively simpler and it has higher generalization ability when being applied to different datasets.

4.3 Methodology

Without the attention mechanism, the CNNs give indifferent treatments to all spatial positions. As the information embedded by the feature responses assumes different significance levels, it is sensible and yet necessary to value those that carry prominent vehicle traits and regulate the influence of those that are less meaningful through an attention mechanism. In this regard, SAM is proposed to coordinate the contribution of various spatial positions based on their underlying information. Meanwhile, powered by MHSA, it also aspires to learn superior and holistic features through the global receptive field.

4.3.1 Spatial Attention Module

SAM is inspired by MHSA in the transformer architecture (Vaswani et al., 2017). MHSA is first applied in Natural Language Processing (NLP) for machine translation tasks and its advent has since challenged the status quo of recurrent neural networks. It allows parallel computation and the modeling of long-range dependency. Motivated by the success of MHSA in NLP, various research issues are steered toward the application of MHSA in the image classification domain. Vision Transformer (ViT) (Dosovitskiy et al., 2020) and Perceiver (Jaegle et al., 2021) were proposed recently and they render comparable classification results against CNNs. Nevertheless, the lack of inductive bias lands the transformer at a disadvantage, especially in the low data regime.

In response to this, this chapter is dedicated to augmenting the understanding of CNN at the global level through the incorporation of SAM. Specifically, SAM treats the feature responses from the top-level feature maps, which are known as spatial positions hereafter, based on the significance of underlying information. Using MHSA as the core building

block, it allocates higher weightage to the spatial positions corresponding to crucial vehicle parts guided by the attention matrices. In particular, the scaled dot-product attention is employed to deduce the attention matrices that quantify the correlation of the spatial positions. To promote diversified learning, the attention matrices are computed in multiple feature spaces via different attention heads and they are eventually used to scale the feature responses appropriately. Generally, SAM will be inserted after the last feature maps of the backbone CNN and the underlying operations are depicted in Figure 4.1.

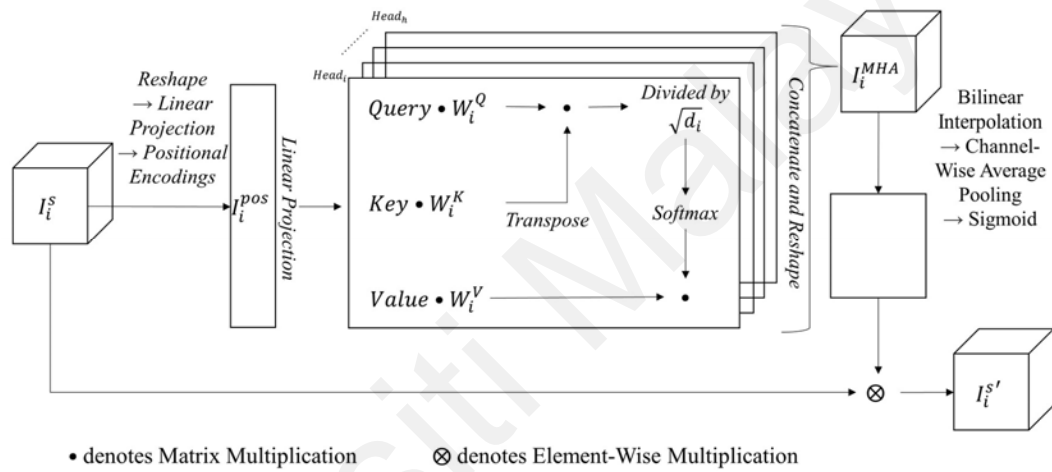


Figure 4.1: Spatial Attention Module

4.3.1.1 Preprocessing of Feature Maps

As MHA works with one-dimensional input, given the input feature maps $I_i^s \in \mathbb{R}^{H \times W \times C}$ where H is height, W is width, C is the number of channels and s is layer index, I_i^s is first reshaped into $I_i^{patch} = \{I_{i,1}^{patch}, I_{i,2}^{patch}, \dots, I_{i,P}^{patch}\} \in \mathbb{R}^{P \times (k \times k \times C)}$ where k is the patch size and $P = (H \times W)/(k \times k)$ is the number of patches.

Subsequently, each image patch is linearly projected such that $I_i^{proj} = \{I_{i,1}^{proj}, I_{i,2}^{proj}, \dots, I_{i,P}^{proj}\} \in \mathbb{R}^{P \times (k \times k \times C)}$ where $I_{i,p}^{proj} = I_{i,p}^{patch} W_{proj}$, p is the patch index and $W_{proj} \in \mathbb{R}^{(k \times k \times C) \times (k \times k \times C)}$.

4.3.1.2 Injection of Positional Information

The position of image patches is important in computing the spatial relationship. However, MHSA does not account for positional differences as the attention operation is carried out in parallel. Being permutation invariant makes MHSA less competitive in modeling highly structured data like images (Bello et al., 2019) and injecting the positional information into the image patches can bring MHSA more clues regarding the object structures. In SAM, positional encodings are chosen as a means to incorporate positional information and the decision is justified in Chapter 4.4.3.1.

Positional encodings inject positional information into I_i^{proj} using sine and cosine functions. The benefit of positional encodings is no training parameters are involved. The operation to generate positional encodings $P_{enc} \in \mathbb{R}^{P \times (k \times k \times C)}$ is as follows:

$$P_{enc}(p, 2t) = \sin\left(\frac{p}{10,000^{\frac{2t}{T}}}\right) \quad (4.1)$$

$$P_{enc}(p, 2t + 1) = \cos\left(\frac{p}{10,000^{\frac{2t}{T}}}\right) \quad (4.2)$$

where $t \in \{0, 1, \dots, (k \times k \times C - 2)/2\}$ and $T = k \times k \times C$. P_{enc} is then added to I_i^{proj} such that $I_i^{pos} = I_i^{proj} + P_{enc}$.

4.3.1.3 Multi-Head Self-Attention

MHSA computes the attention distribution among I_i^{pos} by using query Q , key K and value V where $Q, K, V \in \mathbb{R}^{P \times (k \times k \times C)}$. In self-attention, Q, K and V are essentially I_i^{pos} . As stated by Vaswani et al. (2017), instead of using single-head, it is advantageous to adopt multi-head by linearly projecting Q, K and V for Hd number of times where Hd is the number of heads. MHSA performs the attention computation among Q, K, V matrices

that have been linearly projected into different subspaces. In particular, scaled dot-product attention is calculated in which the output is the weighted sum of V and weight is the degree of compatibility between Q and K . Mathematically, a single-head self-attention operation for $Head_i$ is represented as follows:

$$Head_i = Softmax \left[\frac{QW_i^Q (KW_i^K)^T}{\sqrt{d_i}} \right] (VW_i^V) \in \mathbb{R}^{P \times d_i} \quad (4.3)$$

where $d_i = (k \times k \times C)/Hd$ is the dimension of linearly projected Q , K and V per head, $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{(k \times k \times C) \times d_i}$ are projection matrices for Q , K and V , respectively. Upon computing scaled dot-product attention, all $Head_i$ are concatenated and reshaped into a tensor of shape $(P, k \times k \times C)$ before being sent to the final projection layer.

$$I_i^{MHA} = Reshape([Head_1, Head_2, \dots, Head_{Hd}])W^O \quad (4.4)$$

where $[\bullet]$ is the concatenate operation, $W^O \in \mathbb{R}^{(k \times k \times C) \times (k \times k \times C)}$ and $I_i^{MHA} \in \mathbb{R}^{P \times (k \times k \times C)}$.

4.3.1.4 Recalibration of Feature Maps

In this step, the learned attention information is embedded back into I_i^S . As the output of MHSA is one-dimensional, I_i^{MHA} is reshaped back into two dimensions such that $I_i^{MHA'} = Reshape(I_i^{MHA}) \in \mathbb{R}^{\sqrt{P} \times \sqrt{P} \times (k \times k \times C)}$. When k is larger than 1, it causes \sqrt{P} to be smaller than H and W and hence bilinear interpolation is carried out so that the resultant dimension matches that of I_i^S .

$$I_i^{MHA'} = BilinearInterpolation \left(Reshape(I_i^{MHA}) \right) \quad (4.5)$$

It is important to note that bilinear interpolation is optional and it is needed only when $k \neq 1$. Subsequently, the channel information of $I_i^{MHA'}$ is aggregated using channel-wise

mean operation and the feature responses are kept within 0 to 1 by applying the sigmoid function. Finally, the attention information which contains the spatial information is incorporated into I_i^s by

$$I_i^{s'} = I_i^s \otimes \sigma \left(\text{Mean}(I_i^{MHA'}) \right) \in \mathbb{R}^{H \times W \times C} \quad (4.6)$$

where $\text{Mean}(\bullet)$ is channel-wise mean, \otimes is element-wise multiplication and σ is the sigmoid function.

4.3.2 Spatially Recalibrated Convolutional Neural Network

In this chapter, most of the experiments on SAM are carried out using CaffeNet (Jia et al., 2014) due to its relatively shallow architecture as compared to VGG (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2015) and ResNet (He et al., 2016). Shallow architecture results in lower floating point operations (FLOPs) and hence shorter model training time.

The architecture of CaffeNet is depicted in Figure 4.2. To reduce the number of parameters, group convolution is implemented at Conv2, Conv4 and Conv5 layers. An additional benefit of doing so is that the learned features are more diversified as the channel information is orthogonal between the groups. To introduce non-linearity to the network, the rectified linear unit (ReLU) is opted as the activation function. Maximum pooling is chosen to retain the most salient feature within the pooling kernels. Immediately after the pooling layer, LRN normalizes the feature responses within the neighboring channels. All LRN layers have bias $b = 2$, alpha $\alpha = 10^{-4}$, beta $\beta = 0.75$ and radius $r = 5$. After a series of convolution operations, 3-layer fully connected layers are attached to CaffeNet as the classification head. To increase the sensitivity of CaffeNet towards spatial importance, SAM is embedded after the LRN3 layer since the feature map size at this stage is the smallest and richest in information.

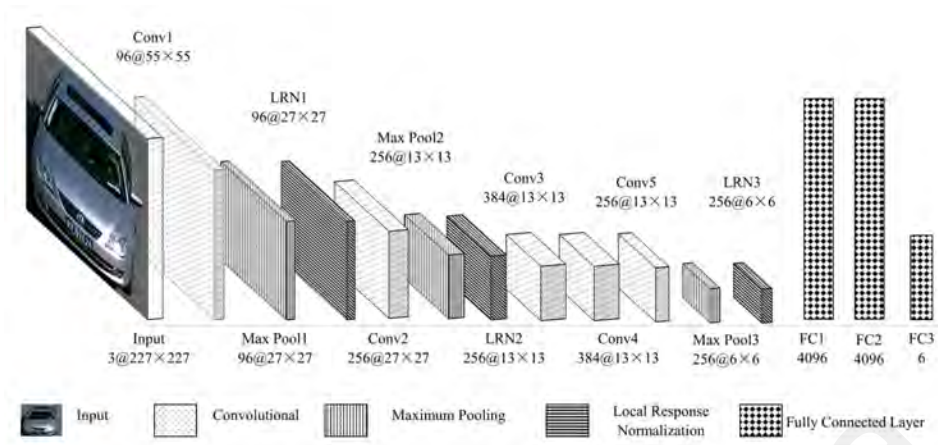


Figure 4.2: CaffeNet

Apart from CaffeNet, SAM is also tested with AlexNet (Krizhevsky, 2014), VGG16 (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2015) and ResNet50 (He et al., 2016). Specifically, SAM is inserted into AlexNet and VGG16 before the first fully connected layer. As for GoogLeNet and ResNet50, it is positioned before the global average pooling (GAP) layer.

4.4 Experiments

4.4.1 Datasets

The experiment for VTR is carried out using the BIT-Vehicle dataset (Dong et al., 2015). It consists of 9,850 images of different viewpoints captured by surveillance cameras and they are subject to external disturbances such as lighting variations, scaling and rotation, etc. This serves as a good reference point to examine the robustness of the proposed framework. Since some images contain more than one vehicle and to suit the vehicle recognition task, the provided annotations are utilized to segment individual vehicles. The class distribution among SUV, sedan, microbus, minivan, truck and bus are 1,372, 5,776, 860, 467, 820 and 555, respectively. Figure 4.3 shows some sample images from the BIT-Vehicle dataset.



Figure 4.3: Sample Images from BIT-Vehicle Dataset

Aside from using the full BIT-Vehicle dataset, 400 images are randomly sampled from each class of which 200 are used for training and 200 for testing to produce a balanced dataset. In other words, this subset contains a total of 2,400 images and the ratio of training to testing images is 50:50. It is worth noting that the performance of SAM is reported based on this subset unless stated otherwise.

To ensure SAM is highly generalizable, the framework is validated further on two publicly available datasets, namely Stanford Cars (Krause et al., 2013) and Web-Nature Comprehensive Cars (CompCarsWeb) (Yang et al., 2015). The particulars of these datasets are shown in Table 4.1. The labels for Stanford Cars are pickup, convertible, sports car, hatchback, MPV, sedan, SUV, minibus and wagon whereas CompCarsWeb has fastback, hardtop and crossover as additional classes.

Table 4.1: Statistics of Stanford Cars and CompCarsWeb

Dataset	Train/ Test	#Classes
Stanford Cars (Krause et al., 2013)	8,144/ 8,041	9
CompCarsWeb (Yang et al., 2015)	36,456/ 15,627	12

4.4.2 Implementation Details

Firstly, the images are resized to 224×224 before being normalized based on the ImageNet dataset. CaffeNet is then initialized with weights pretrained on the ImageNet dataset. For SAM, $k = 1$ and $Hd = 8$ are set. On the BIT-Vehicle dataset, CaffeNet-SAM is fine-tuned based on cross entropy loss using Adam (Kingma & Ba, 2014) for 50 epochs. The learning rate is set as 1e-4 for the first 25 epochs and it is decayed by a factor of 10 at 26th epoch. For Stanford Cars and CompCarsWeb datasets, the network is trained for 90 epochs since they are larger in quantity. Stochastic Gradient Descent (SGD) is chosen as the optimizer with 0.9 momentum and 5e-4 weight decay. The learning rate is set as 0.01 and decays by a factor of 10 for every 40 epochs. Random cropping is also performed to prevent overfitting during training.

The experiment is performed on a machine with the specification of Intel Core i7-9750H 2.6GHz, 32GB RAM and NVIDIA Quadro T1000 4GB video memory.

4.5 Results & Discussions

4.5.1 Quantitative Analysis

The proposed framework is benchmarked against the existing works on the subset of the BIT-Vehicle dataset (Dong et al., 2015) in Table 4.2 based on the metric calculated below

$$Accuracy = \frac{TP}{N_{Test}} \times 100 \quad (4.7)$$

where TP is the true prediction count and N_{Test} is the testing image count.

Table 4.2: Performance Benchmarking on BIT-Vehicle (Subset)

Reference	Train/ Test	Accuracy
2D Deep Boltzmann Machine (Santos et al., 2017)	1,200/ 1,200	80.6%
Haar Cascade Classifier (Baser & Altun, 2016)	-	81.8%
SF (Dong et al., 2015)	1,200/ 1,200	86.8%

Table 4.2 Performance Benchmarking on BIT-Vehicle (Subset), continued

Reference	Train/ Test	Accuracy
SLFL (Dong et al., 2015)	1,200/ 1,200	88.1%
PCN-Softmax (Soon et al., 2020)	1,200/ 1,200	88.5%
KNNPC + DSRC (Sun et al., 2017)	1,200/ 1,200	90.1%
DPM + SVM (Bai et al., 2018)	1,200/ 1,200	91.1%
CaffeNet-SAM	1,200/ 1,200	94.2%
ViT-B (Dosovitskiy et al., 2020)	1,200/ 1,200	94.5%
APINet (Zhuang et al., 2020)	1,200/ 1,200	94.8%
SwinV2-T (Ze Liu et al., 2022)	1,200/ 1,200	94.9%
LRAU (Boukerche & Ma, 2021)	1,200/ 1,200	95.1%
APCNN (Ding et al., 2021)	1,200/ 1,200	95.3%
HERBS (Chou et al., 2023)	1,200/ 1,200	95.3%
CMAL (D. Liu et al., 2023)	1,200/ 1,200	95.4%
ResNet50-SAM	1,200/ 1,200	95.4%

CaffeNet-SAM reports 94.2% accuracy whereas ResNet50-SAM tops the ranking with 95.4% accuracy. Santos *et al.*'s work (Santos et al., 2017) which feeds the images projected by 2D Linear Discriminant Analysis into the Boltzmann Machine reports 80.6% accuracy. Baser and Altun (2016) use the Haar Cascade Classifier to both detect and classify vehicle types and achieve 81.8% accuracy. Dong et al. (2015) use a semi-supervised method to learn the convolution kernels of CNN. Specifically, they experiment with Sparse Filtering (SF) (Ngiam et al., 2011) to optimize the convolution kernels for sparsity. SLFL method is an improvisation over SF by taking reconstruction error, sparsity and manifold assumption into account (Dong et al., 2015). Accuracies of 86.8% and 88.1% are reported by SF and SLFL, respectively. Soon et al. (2020) use PCA to learn the convolution filters and they report 88.5% accuracy. Although both Dong et al. (2015) and Soon et al. (2020) achieve remarkable accuracy, their frameworks are not trainable end-to-end as the convolution kernels need to be optimized beforehand separately. Work by Sun et al. (2017) delivers 90.1% accuracy. Their network is trained to first classify the vehicles into heavy or light before recognizing the types and hence additional labels are required. Bai *et al.*'s work (Bai et al., 2018) with 91.1% accuracy is required to produce deformable part models (DPM) for each vehicle type and this may

impact the requirement for real-time inferencing when the number of vehicle types increases.

Additionally, more works, especially those from the attention mechanism domain, are included for comparison purposes. Since these works do not report the performances on the BIT-Vehicle dataset originally, the training process as elucidated in Chapter 4.3.2 is performed and the results are reported in Table 4.2. ViT-B (Dosovitskiy et al., 2020) which has around 86M parameters delivers 94.5% accuracy. It is criticized for the inability to encapsulate information from all image patches into the class token (Kang et al., 2022; Touvron et al., 2021; Yuan et al., 2021). APINet (Zhuang et al., 2020) which leverages pairwise contrastive clues achieves 94.8% accuracy. It is hypothesized that better performance can be achieved by customizing a dataset-specific pair construction strategy for training. SwinV2-T (Ze Liu et al., 2022) is a transformer network that employs a shifting windowing scheme for MHSA and it reports 94.9% accuracy. Although the devised MHSA has linear complexity, the capability to model global dependency is compromised.

LRAU (Boukerche & Ma, 2021) outperforms the shallow architecture i.e. CaffeNet-SAM but its accuracy is 0.3% lower than ResNet50-SAM. APCNN (Ding et al., 2021) reports 95.3% accuracy based on multi-level classification heads. It is reckoned that attaching a classification head at early convolution layers leads to contradiction in feature learning as the low-level feature maps have to learn both high-level semantic information and low-level fine-grained information at the same time. ResNet50-based High-temperature Refinement and Background Suppression (HERBS) (Chou et al., 2023) is as competitive as APCNN. With the help of the selector module, it identifies the salient feature responses from various pyramid levels and channels them into a Graph Convolutional Network-based combiner module for cross-granularity information

exchange. For Cross-Layer Mutual Attention Learning (CMAL) (D. Liu et al., 2023), it trains multiple classification experts that first segment the vehicle from the image in a weakly supervised manner through feature maps binarization before performing the classification. Building upon TResNet-L (Ridnik et al., 2021a), CMAL has parameters as many as 63.2M. Although ResNet50-SAM is smaller than CMAL by close to 20M, it renders the same performance level.

The proposed framework is also examined on a larger image pool. Since the complete set of the BIT-Vehicle dataset is imbalanced, the macro average recall (MARecall) and accuracy figures are both attached in Table 4.3. The MARecall is calculated as follows:

$$\text{Macro Average Recall} = \frac{\sum_l^L \frac{TP_l}{N_l}}{L} \times 100 \quad (4.8)$$

where TP_l and N_l are true prediction count and total image count in class l . It is worth noting that MARecall is equivalent to accuracy when the images from each class have equal proportions.

Table 4.3: Performance Benchmarking on BIT-Vehicle (Full)

Reference	Train/ Test	MA Recall	Accuracy
Inception-v3 (P. Liu et al., 2021)	7,880/ 1,970	Unreported	97.1%
Stereo-Vision Based Model (Derrouz et al., 2019)	7,895/ 1,955	Unreported	95.2%
CNN (Roecker et al., 2018)	29,760/ 852	93.9%	93.9%
Super Leaner Ensemble (Hedeya et al., 2020)	8,039/ 2,014	96.8%	97.6%
CaffeNet-SAM	7,881/ 1,969	95.4%	97.4%
ViT-B (Dosovitskiy et al., 2020)	7,881/ 1,969	95.9%	97.7%
APINet (Zhuang et al., 2020)	7,881/ 1,969	96.0%	97.6%
SwinV2-T (Ze Liu et al., 2022)	7,881/ 1,969	96.1%	97.6%
HERBS (Chou et al., 2023)	7,881/ 1,969	95.8%	97.6%
LRAU (Boukerche & Ma, 2021)	7,881/ 1,969	96.2%	97.7%
APCNN (Ding et al., 2021)	7,881/ 1,969	96.4%	97.8%
CMAL (D. Liu et al., 2023)	7,881/ 1,969	96.6%	98.1%
ResNet50-SAM	7,881/ 1,969	96.9%	98.2%

P. Liu et al. (2021) report 97.1% accuracy based on Inception-v3 (Szegedy et al., 2016). Derrouz et al. (2019) report 95.2% accuracy using vehicle dimensions and HOG as features. Their stereo vision-based work requires two cameras and this results in higher installation costs. A novel CNN introduced by Roecker et al. (2018) reports 93.9% recall and 93.9% accuracy. Hedeya et al. (2020) propose a super-learner ensemble technique based on Xception (Chollet, 2017) and DenseNet.(Huang et al., 2017). In particular, the logits of two deep learning models are merged via a fully connected layer and 96.8% recall and 97.6% accuracy are reported. The transformer-based ViT-B (Dosovitskiy et al., 2020) achieves 95.9% recall and 97.7% accuracy. APINet (Zhuang et al., 2020) and SwinV2-T (Ze Liu et al., 2022) are on par by reporting 97.6% accuracy. For HERBS (Chou et al., 2023), a drop in ranking is seen as its performance is poorer than LRAU (Boukerche & Ma, 2021) and APCNN (Ding et al., 2021) in terms of accuracy. This is caused by the selector module which constrains the feature representation learning from the most discriminative responses. Consequently, other complementary visual cues are forgone and the embeddings become less diverse. APCNN consistently outflanks LRAU by claiming 96.4% recall and 97.8% accuracy. CMAL (D. Liu et al., 2023) remains the best network after ResNet50-SAM. It is worth noting that CMAL has high computational costs which stand at 14.04 GFLOPs since 5 forward passes are required to deduce a superior vehicle segmentation mask. For the proposed framework, CaffeNet-SAM achieves 95.4% recall and 97.4% accuracy. ResNet50-SAM achieves the best performance with 96.9% recall and 98.2% accuracy.

As both Stanford Cars (Krause et al., 2013) and CompCarsWeb (Yang et al., 2015) datasets are considerably large, a deep CNN i.e. ResNet50 (He et al., 2016) is utilized here. The results are tabulated in Table 4.4. The efficacy of SAM on these datasets is demonstrated where it identifies and pays more attention to critical spatial positions to render better classification performance than the baseline. ResNet50-SAM achieves

84.5% and 96.0% accuracy on Stanford Cars and CompCarsWeb. The improvement brought by SAM is 1.6% and 3.1%, respectively.

Table 4.4: Performance Benchmarking on Stanford Cars and CompCarsWeb

Dataset	Model	Accuracy
Stanford Cars (Krause et al., 2013)	ResNet50	82.9%
	ResNet50-SAM	84.5%
CompCarsWeb (Yang et al., 2015)	ResNet50	92.9%
	ResNet50-SAM	96.0%

To render a holistic evaluation, a comparison between the vanilla CNN and post-SAM insertion, as represented by bold text, in terms of the number of parameters, FLOPs and inference speed is tabulated in Table 4.5. Incorporating SAM brings negligible effect on the network size and computational cost to CaffeNet. This is attributed to the low channel count of the top-level features i.e. 256 channels. On the contrary, incorporating SAM almost doubles the size of ResNet50 and this is due to the top-level feature maps that have 2,048 channels. The high channel count also translates to a larger increment in FLOPs as compared to CaffeNet and the increment approximates 1 GFLOPs, Nevertheless, ResNet50-SAM is still considered moderate in size.

For inference time, CaffeNet-SAM outflanks the framework proposed by Soon et al. (2020) by 7 times, which is 1 ms against 7 ms whereas the inference time of ResNet50-SAM is 10 ms. Nevertheless, this should just serve as a reference as the machine specification is not considered. The inference time for the rest of the works is unreported.

Table 4.5: Computational Complexity of SAM

Model	#Params (M)	GFLOPs	Inference Time per Frame (ms)
CaffeNet	57.0	0.7	1
CaffeNet-SAM	57.3	0.7	1
ResNet50	23.5	4.1	8
ResNet50-SAM	44.5	5.2	10

4.5.2 Qualitative Analysis

To examine the learned deep features, T-distributed Stochastic Neighbour Embedding (TSNE) (van der Maaten & Hinton, 2008) is applied to project the 4,096-dimensional features extracted from the penultimate fully connected layer of CaffeNet-SAM into two dimensions for visual inspection. The TSNE plot of the deep features is shown in Figure 4.4 and each point is labeled with the actual class label.

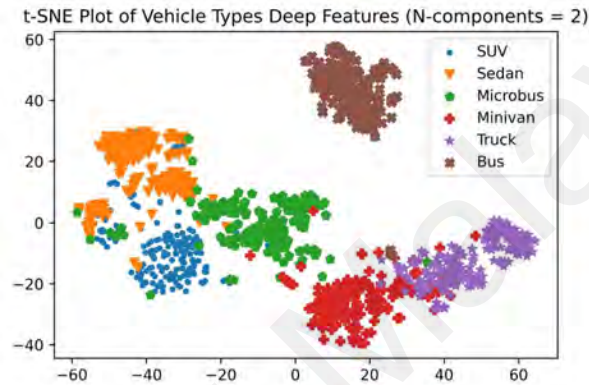


Figure 4.4: TSNE Plot of Deep Features from CaffeNet-SAM

It is observed that the bus is distinctive from the rest of the classes as it forms an independent cluster far away from others. This is due to its prominent large vehicle size and the rigid rectangular shape which present a large visual difference from the rest. As for SUV, sedan and microbus, separation is less clear but the three clusters are still clearly visible. Besides, it is noticed that minivan and truck share a higher inter-class similarity as compared to other classes due to similar vehicle fronts.

4.5.3 Ablation Study

For a more detailed examination of how each design parameter affects the performance of SAM, an ablation study is performed on CaffeNet-SAM using the subset of the BIT-Vehicle dataset.

4.5.3.1 Effect of Positional Information On SAM

By nature, MHSA takes no notice of the order of image patches (Vaswani et al., 2017). However, the sequence information is important as it avoids permutation equivariance and reinforces the contextual understanding of an object. In SAM, the positional information is explicitly inserted using positional encodings. An alternative way to inject the positional information is also experimented. Specifically, one-dimensional learnable positional embeddings (Dosovitskiy et al., 2020) is adopted and it is initialized using uniform distribution $U(-0.05, 0.05)$.

Table 4.6 shows the performance of SAM using different approaches to inject the positional information. It is observed that positional information is important in SAM and positional encodings performs better than positional embeddings by 0.4%. Comparing positional encodings and embeddings, the sinusoidal waveform utilized by positional encodings retains the inter-patch relativity and thus allows the spatial structure of the vehicle to enrich the feature representation. On the contrary, positional embeddings encodes only the absolute positional information and it fails to model the patch-to-patch relationship. Another additional benefit of positional encodings is it reduces the trainable parameters of the network since the positional information is calculated explicitly rather than being learned during the training process.

Table 4.6: Effect of Positional Information on SAM

Method to Inject Positional Information	Accuracy
W/O Injecting Positional Information	93.5%
Positional Embeddings (Gehring et al., 2017)	93.8%
Positional Encodings (Vaswani et al., 2017)	94.2%

4.5.3.2 Effect of Patch Size on SAM

In this section, the optimal value for k to restructure the feature maps into patches is examined. Based on the implementation, since CaffeNet performs convolution to achieve

spatial reduction, k is set to be significantly smaller than ViT (Dosovitskiy et al., 2020) i.e. 16 to prevent overboard generalization over a large number of feature responses.

Table 4.7 presents the classification results for different values of k . The results suggest that setting $k = 1$ reaches the highest classification accuracy and the performance declines with increasing patch size. Increasing patch size from 1 to 3 brings 0.9% reduction in accuracy. This is because each feature response on the feature maps produced by the last convolution layers corresponds to large receptive fields. Setting $k > 1$ is detrimental to the feature distinctiveness as the over-integration of vehicle parts results in the loss of fine-grained vehicle cues.

Table 4.7: Effect of Patch Size on CaffeNet-SAM

Patch Size	Accuracy
1	94.2%
2	93.7%
3	93.3%

4.5.3.3 Effect of Number of Heads on SAM

In SAM, multi-head is chosen over single-head self-attention. MHSA provides the flexibility of attending to different subspace representations. Nevertheless, there are no standard methods to determine the optimum number of heads. Hence, it is determined empirically and the results are presented in Figure 4.5.

Conforming to the expectation, MHSA indeed performs better than single-head self-attention. MHSA promotes diversified learning where each attention head models different intricate vehicle parts to improve the feature expressiveness. Furthermore, computing the attention in different subspaces provides better generalization ability and eventually reduces the chances of overfitting. The optimum number of heads for CaffeNet-SAM is 8 in this study. Setting Hd as 4 and 16 delivers the same performance.

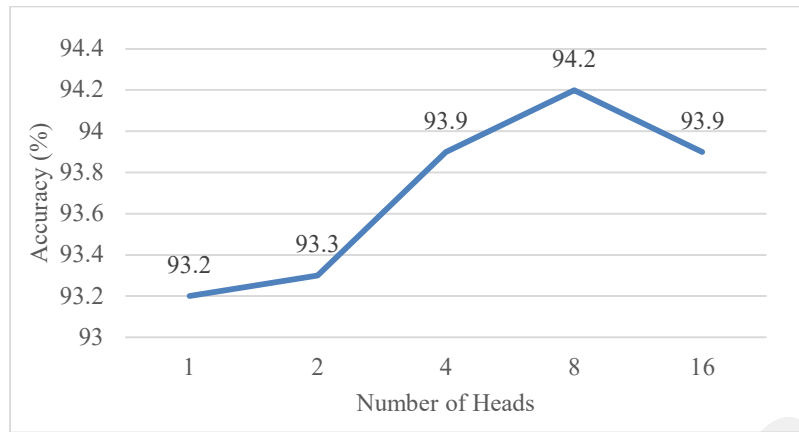


Figure 4.5: Effect of Number of Heads on CaffeNet-SAM

4.5.4 Generalization Study

Figure 4.6 shows the accuracy of different CNNs tested on the subset of the BIT Vehicle dataset. These CNNs are implemented using the open-source PyTorch framework and they are trained in the same fashion as described in Chapter 4.3.2.

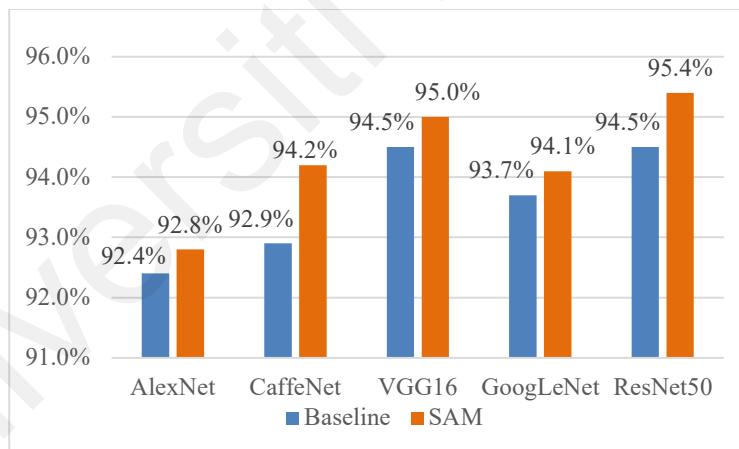


Figure 4.6: Compatibility between the SAM and Existing CNNs on BIT Vehicle (Subset)

Based on the results, it is observed that before incorporating SAM, the lowest accuracy is recorded by the shallowest CNNs, namely AlexNet (Krizhevsky, 2014) and CaffeNet (Jia et al., 2014). This is followed by GoogLeNet (93.7%) (Szegedy et al., 2015) which adopts Inception Module. VGG16 (Simonyan & Zisserman, 2014) which uses fixed-size 3×3 convolution kernels reports 94.5% accuracy. ResNet50 (He et al., 2016) which

implements a skip connection strategy has the largest number of layers among all. It shares the same accuracy with VGG16.

Upon incorporating SAM, all networks show improvement by an average of 0.7%. CaffeNet records the largest leap in accuracy, which is by 1.3% followed by 0.9% of ResNet50. The results indicate that due to the limited receptive field of convolution kernels, the feature maps fail to gain a holistic understanding of the vehicles and hence the learned embeddings are still weak semantically. With SAM, the inter-spatial relationship is computed by MHSA which exerts a global receptive field. The thorough propagation of all spatial information enables the distinctive features to be pinpointed and recalibrated to elevate the classification performance.

4.5.5 Performance of SAM against Existing Attention Modules

As SAM employs the concept of attention, its performance is benchmarked against the existing attention modules. Specifically, comparisons are made against the Squeeze-and-Excitation (SE) block (Hu et al., 2018), Bottleneck Attention Module (BAM) (Park et al., 2018) and CBAM (Woo et al., 2018). These modules are configured as per the optimal value obtained from the respective works.

Table 4.8 compares SE, BAM, CBAM and SAM when integrated with CaffeNet in terms of classification accuracy. SE, BAM and CBAM are inserted after every LRN layer of CaffeNet. In other words, there are a total of three attention blocks being added. As SE, BAM and CBAM are originally proposed based on ResNet, they are also integrated into ResNet50 following the practices of Hu et al. (2018), Park et al. (2018) and Woo et al. (2018).

Table 4.8: Performance Comparisons against Existing Attention Modules on BIT-Vehicle (Subset)

Attention Module	Backbone	
	CaffeNet	ResNet50
SE (Hu et al., 2018)	93.7%	95.0%
BAM (Park et al., 2018)	91.1%	94.9%
CBAM (Woo et al., 2018)	90.8%	94.7%
SAM	94.2%	95.4%

The results show that although SE outperforms BAM and CBAM, it falls behind SAM by an average of 0.5%. Similar to the SE, SAM exploits the inter-channel dependency through linear projection operation. Nevertheless, channel recalibration alone leads to less conclusive results since the spatial context information is not considered. Therefore, following channel recalibration, SAM leverages the scaled dot-product attention to compute the correlation of the spatial positions before scaling the feature responses based on their relative importance. The results unravel that spatial feature refinement based on MHSA is pivotal to discriminating against different vehicle types.

4.6 Conclusion

In this chapter, SAM that treats each spatial position on the feature maps according to the information relevancy is proposed. It places a higher focus on spatial positions that correspond to key vehicle parts and attenuates the insignificant information to better differentiate vehicle types. SAM is fused with CaffeNet and it delivers 97.4% accuracy for 6 vehicle types on the BIT-Vehicle dataset. It takes 1 ms during inference and it is fit for real-time implementation. Integrating SAM into deeper CNN renders even better classification performance where ResNet50-SAM achieves 98.2% accuracy. In addition, SAM leads the state-of-the-art solutions, especially those that originate from the attention domain by a considerable margin. It also exhibits a high generalization ability where it brings improvement over the baseline network by an average of 2.4% accuracy on Stanford Cars and CompCarsWeb datasets.

CHAPTER 5: CROSS-GRANULARITY NETWORK FOR VEHICLE MAKE AND MODEL RECOGNITION

5.1 Introduction

Most of the extant Vehicle Make and Model Recognition (VMMR) solutions share one thing in common, which is they forgo the low-level feature maps and solely rely on the deep features of Convolutional Neural Networks (CNNs) to infer the vehicle models (L. Wang et al., 2022). Although high-level feature maps carry sound global information, the ability to detect interclass differences is dubious as it lacks spatial and structural information about an object. The succor to this is by considering multi-scale features that are highly inclusive where both microscopic and macroscopic information are exploited to empower the network with high differentiation ability.

In this chapter, a Cross-Granularity (CG) module is proposed to amalgamate information from various scale levels in stages. The proposed CG module is inserted into the backbone CNN to extract features across all pyramid levels via convolution and pooling operations. The hierarchical features are subsequently fused through channel-wise pooling to form the final feature maps that enclose the cross-scale vehicle features and they serve as input to the classification head for logit computation. The advantages of the CG module can be expounded from three aspects. Firstly, it leverages dilated convolution to expand the field of views of convolution kernels to derive more extensive features. Secondly, it preserves the granular characteristics of the vehicles from the shallow layers by establishing a direct connection with the deep-layer feature maps instead of passing them from layer to layer. Thirdly, it dynamically aggregates the scale-varying components based on channel significance via 1×1 convolution. The CG module enhances the performance of backbone CNNs where competitive VMMR performances

are reported on several public datasets. The contributions of this chapter are summarized as follows:

- Propose CG module to aggregate multi-scale information seamlessly by adaptively consolidating local and global features to present coarse-to-fine semantic information within a holistic feature map
- Examine the CG module on several public datasets and benchmark its performance against state-of-the-art solutions
- Validate the ability of the CG module in focusing fine-grained details by visualizing the learned features

5.2 Literature Review

VMMR is an important task in Intelligent Transportation System (ITS) and has been actively researched over the years. Different methodologies are proposed and they are composed of feature-based, part-based features, attention mechanisms and multi-scale features.

Feature-based techniques adopt popular feature descriptors such as Histogram of Oriented Gradient (HOG), Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF). Since these raw handcrafted features are susceptible to external disturbances (Chan et al., 2015; Dong et al., 2015; Ge et al., 2017; Huang et al., 2015; S. Li et al., 2018), recent feature-based approaches are paired with feature encoding schemes to boost the robustness. Siddiqui et al. (2016) combined SURF with Bag of Features (BoF) to produce mid-level features. Nazemi et al. (2020) considered locality constraints by encoding dense-SIFT features with Locality-constraint Linear Coding (LLC). Jamil et al. (2020) improvised BoF into Bag of Expressions (BoE) to increase the tolerance against viewpoint variation. Although there is considerable improvement brought by the feature encoding schemes, an optimum feature extractor still mainly accounts for the

classification performance and making the right choice requires a vast amount of experience.

With the advancement in the deep learning domain, CNN has rapidly gained attention in VMMR. Part-based approaches are built upon CNNs. They look for key vehicle parts and eliminate the interference of background information completely. They are easy to train as localization is learned in a weakly supervised manner without using bounding box annotations. Biglari et al. (2017a) used latent Support Vector Machine (SVM) and cascade classifiers to identify and classify the prominent parts. Their solution has an inference speed as low as 0.01 frame per second. Fang et al. (2016) isolated the discriminative vehicle parts by visualizing neuron activity through heatmaps in their Coarse-to-Fine CNN but the localization strategy only works under a single viewpoint. Fu et al. (2017) presented Recurrent Attention-CNN which features an Attention Proposal Network to recurrently zoom into discriminative parts. The end-to-end pipeline requires multiple forward passes and it scales up the computational costs. In Multiscale Attention Windows Network (MAWNet), Ghassemi et al. (2019) demonstrated how distinctive vehicle parts can be localized using the Spatial Transformer Network (STN). Nevertheless, MAWNet is not trainable end-to-end as STN has to undergo a separate training process to ensure convergence of loss function.

Attention modules are normally integrated into the CNNs. Unlike part-based methods, they perform feature enhancement so that the important features are given higher weightage and redundant features are suppressed. Spatially Weighted Pooling (SWP) (Hu et al., 2017) treats each spatial position distinctively according to importance but it is sensitive to changes in viewpoint. The Convolutional Attention Model (ConvAM) (Yu et al., 2020) adopts Long Short-Term Memory (LSTM)-based attention modules which are computationally expensive to refine the feature maps. Attention Pairwise Interaction

Network (APINet) (Zhuang et al., 2020) looks for cross-image interaction but it requires a proper image pair construction strategy for effective learning. Given the recent attention on the transformer architecture, a Fine-Grained Transformer (TransFG) (He et al., 2021) modified from Vision Transformer (ViT) (Dosovitskiy et al., 2020) is proposed. TransFG uses only the most informative image token from each attention head for classification but this poses the risk of overfitting.

To compute multi-scale features, G. Wang et al. (2021) modified the bottleneck block of ResNet50 (He et al., 2016) by replacing the 3×3 convolution with Pyramid Convolution (PyConv) that runs convolution kernels of different sizes. Their proposal also includes a multi-attention module that implements channel and spatial attention to recalibrate feature maps based on saliency. Similarly, L. Wang et al. (2022) constructed the last convolutional block (CB) of ResNet34 (He et al., 2016) with Parallel Convolutional Block (PCB). Each PCB executes multi-stream convolutions that comprise 1×1 and 3×3 convolution kernels. The resultant feature maps from all PCB blocks are subsequently consolidated via Multilayers Feature Fusion (MFF) through matrix outer product and normalization. In Feature Integration and Feature Fusion Network (FIFFNet), P. Wang et al. (2022) preserved the global and local components by aggregating the feature maps from the last two layers through summation. Although G. Wang et al. (2021), L. Wang et al. (2022) and P. Wang et al. (2022) improved the classification performance, the final feature maps are still deprived of local subtle details since the cross-scale feature generation only involves the top pyramid levels. On the contrary, the CG module ingests multiple scale-varying components by augmenting lateral connections from various image pyramids.

Progressive Multi-Granularity (PMG) network (Du et al., 2020) exploits the multi-scale features by examining the scale-specific feature maps independently to arrive at the

final prediction. The absence of cross-layer feature interaction causes performance degradation. The CG module overcomes this weakness by concatenating the pyramidal features and employing 1×1 convolution to realize end-to-end information transmission. A partial cross-layer interaction is implemented in the Feature Covariance Attention (FCA) module (Jung et al., 2017) where the information exchange is limited within the adjacent scales. The design is inappropriate as it causes the large-scale components to dominate over the small-scale components. The CG module prevents this by considering feature maps from all hierarchies in one go and synthesizing them based on their relative importance throughout the training process. Feature Pyramid Network (FPN) (Lin et al., 2017a) is a notorious technique in synthesizing multi-scale features but its relatively simple feature fusion method limits the feature representation learning process. In the CG module, the upsampling and addition operations utilized by FPN are eradicated and substituted with a dynamic feature merging mechanism powered by 1×1 convolution that effectively distills both coarse- and fine-grained components.

5.3 Methodology

5.3.1 Cross-Granularity Module

The architecture of the CG module is portrayed in Figure 5.1. Its motivation is to exploit the inter-scale relationship of pyramidal features to generate scale-aware representations so that the low-level subtle details and high-level semantic information are leveraged concurrently for fine-grained classification.

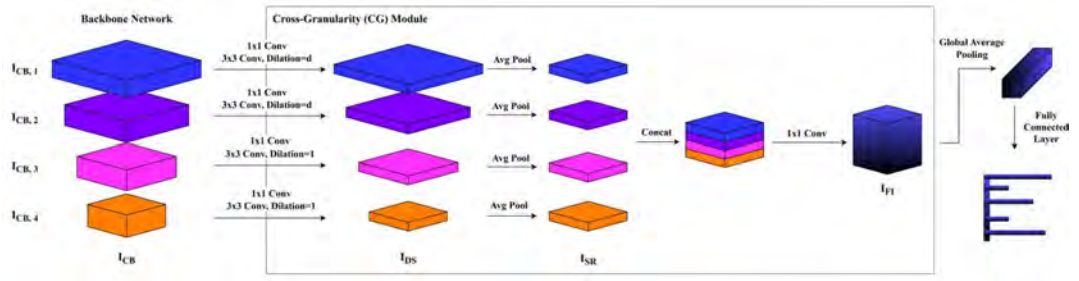


Figure 5.1: Cross-Granularity Network

By nature, given an input image $I_i \in \mathbb{R}^{H \times W \times C}$ where H, W and C represent height, width, and number of channels, a CNN produces feature maps with different spatial resolutions and depths hierarchically through its CBs. The output feature maps from each CB can be generalized as $I_{CB,i} \in \mathbb{R}^{H_{CB,i} \times W_{CB,i} \times C_{CB,i}}$ and the number of CBs is designed to be 4 for most networks.

The CG module is composed of two stages which are the Feature Extraction (FE) and Feature Integration (FI) stages. The FE stage is mainly responsible for standardizing the resolutions and depth for each $I_{CB,i}$ whereas the FI stage will consolidate them to synthesize cross-grained feature embeddings. It is important to note that only the $I_{CB,i}$ from the last N_{CB} CB will be ingested into the CG module. Setting an optimum N_{CB} is crucial as it regulates the amount of shallow-layer information to be taken into account. Under-involvement of low-level features causes the loss of subtle visual traits whereas the contrary adversely affects the classification performance due to the existence of noisy information.

Essentially, the FE stage first operates 1×1 convolution to standardize the feature map depth. Since it is construed that feature maps from all levels are equally important and none of them should be over-represented over the others, the depth is set as

$$C_{FE} = \left\lceil \frac{C_{CB,4}}{N_{CB}} \right\rceil \quad (5.1)$$

The 3×3 convolution then follows suit to perform further feature extraction with a dilation rate $d > 1$ for $\{I_{CB,i}\}_{i=1}^2$. The reason is that given large feature maps produced by early CBs, 3×3 convolution kernels have a small receptive field size. In addition, since the discriminative vehicle parts are usually fragmented and scattered over the entire objects, this makes the aggregation of significant cues over a wide region infeasible. Consequently, the deduced features are suboptimal. To mitigate this, dilated convolution is adopted to increase the receptive fields and extract more holistic features by considering a broader area while keeping the number of trainable parameters of the convolution layer the same. These operations produce $I_{DS,i} \in \mathbb{R}^{H_{CB,i} \times W_{CB,i} \times C_{FE}}$ and they are represented as

$$I_{DS,i} = Conv_{3 \times 3, C_{FE}, d} \left(Conv_{1 \times 1, C_{FE}} (I_{CB,i}) \right) \quad (5.2)$$

where $Conv_{1 \times 1, C_{FE}}$ is 1×1 convolution with C_{FE} output channels and $Conv_{3 \times 3, C_{FE}, d}$ is 3×3 convolution with C_{FE} output channels and dilation rate d . For each convolution layer, it is followed by batch normalization (BN) and rectified linear unit (ReLU).

Upon reducing the depth, $I_{DS,i}$ is downsized into smaller resolutions. To keep the FLOPs minimal, the spatial resolutions of $I_{DS,i}$ is standardized according to the output of the last CB through average pooling (AP). AP is chosen instead of maximum pooling because it can aggregate the information within the kernel well whereas maximum pooling which pays attention only to the salient neuron activity results in information loss. The spatial reduction operation is denoted as

$$I_{SR,i} = AP(I_{DS,i}) \quad (5.3)$$

where $I_{SR,i} \in \mathbb{R}^{H_{CB,A} \times W_{CB,A} \times C_{FE}}$ is the resultant feature maps.

After standardizing the feature maps, the subsequent challenge is how best these features which carry different levels of semantic information can be fused. The objective is to integrate the multi-level feature maps seamlessly while being capable of learning the non-linear interaction among them. To be exact, all $I_{SR,i}$ are concatenated along the channel axis and 1×1 convolution is applied as cross-channel pooling.

$$I_{FI} = Conv_{1 \times 1, C_{FI}} \left(\left[\{I_{SR,i}\}_{4-N_{CB}+1}^4 \right] \right) \quad (5.4)$$

where $[\bullet]$ is concatenation along the channel axis and $C_{FI} = C_{FE} \times N_{CB}$. BN and ReLU are performed after 1×1 convolution. Lastly, I_{FI} is sent into the classification head to produce the logits. The number of fully connected layers N_{FC} in the classification head assumes one of the configurations illustrated in Table 5.1 to suit the dataset of different complexity levels where GAP is the global average pooling and $FC_{C_{FE}}$ and FC_L are the fully connected layers with C_{FE} and L output neurons.

Table 5.1: Configuration of Classification Head

N_{FC}	Layer
1	$GAP \rightarrow BN \rightarrow FC_L$
2	$GAP \rightarrow BN \rightarrow FC_{C_{FE}} \rightarrow BN \rightarrow ReLU \rightarrow FC_L$

5.3.2 Cross-Granularity Network

The CG module is a highly modular component that can be inserted into any CNNs. To manifest its ability in elevating the feature perception ability of the backbone network, three infamous CNNs are used in this experiment during the performance validation process. Generally, a backbone network that is integrated with the CG module is denoted as CG Network (CGNet).

Inceptionv3 (Szegedy et al., 2016) is mainly constructed from the Inception module. The Inception module implements factorized convolution and asymmetric convolution to

learn distinctive features with an adequate number of parameters. For an effective learning process, an auxiliary classifier is attached at the intermediate layer to propagate the gradients down the network. Based on $299 \times 299 \times 3$ I_i , feature maps $71 \times 71 \times 192$, $35 \times 35 \times 288$, $17 \times 17 \times 768$ and $8 \times 8 \times 2048$ are identified as $\{I_{CB,i}\}_{i=1}^4$.

ResNet50 (He et al., 2016) is a network that performs identity mapping using shortcut connections to facilitate the gradient flow. It alleviates the vanishing gradient problem and allows the network to learn effectively even with a high number of layers. Specifically, ResNet50 which contains 50 convolution layers is used due to its exemplary performance exhibited in numerous works. Based on $224 \times 224 \times 3$ I_i , $\{I_{CB,i}\}_{i=1}^4$ are the feature maps produced by every bottleneck block.

The dense connection of DenseNet169 (Huang et al., 2017) empowers seamless information sharing across layers. With the densely connected architecture, a network that is significantly deeper than ResNet50 is developed. Based on $224 \times 224 \times 3$ I_i , $\{I_{CB,i}\}_{i=1}^4$ are the feature maps produced by every dense block and transition layer.

5.3.3 Loss Function

A loss function is used to measure how well the network models the training data. Cross entropy loss is adopted throughout the experiment. Label smoothing (Szegedy et al., 2016) is also adopted for some of the experimental datasets as the regularization technique. It introduces uniformly distributed noise to the ground truth labels to prevent the model from overfitting. It also mitigates the effect of wrongly labeled data by being less skeptical about getting the correct prediction. Cross entropy loss with label smoothing is generalized as

$$Loss = -\frac{1}{N_{Train}} \sum_{i=1}^{N_{Train}} \sum_{l=1}^L (1 - \varepsilon) y_{i,l} \log p_{i,l} + \frac{\varepsilon}{(L - 1)} \log p_{i,l} \quad (5.5)$$

where \mathcal{E} is the label smoothing coefficient, $y_{i,l}$ is one-hot encoded ground truth, $p_{i,l}$ is the output distribution predicted by the model and N_{Train} is the number of training samples.

5.4 Experiments

5.4.1 Datasets

A total of 4 public fine-grained vehicle datasets are used in the experiments. Figure 5.2 shows some sample images. Web-Nature Comprehensive Cars (CompCarsWeb) (Yang et al., 2015) consists of web-collected vehicle images that cover 431 vehicle models captured from various viewpoints. For a fair comparison with existing works, the provided train-test split is adopted and it results in 36,456 training and 15,627 testing images, respectively.



Figure 5.2: Sample Images from CompCarsWeb, Stanford Cars, Car-FG3K and CompCarsSV

Stanford Cars (Krause et al., 2013) is another well-known dataset used in fine-grained classification tasks. Similar to CompCarsWeb, it contains multi-view vehicle images scraped online from 196 vehicle models. This dataset is slightly challenging as the train-test split ratio is even, with 8,144 train and 8,014 test images.

Car-FG3K (Wu et al., 2022) is a dataset published recently that contains 1,892 vehicle models. Despite having a huge number of classes, the training images are not abundant. The images are captured from various viewpoints. No bounding box annotations are provided to eliminate the background noise. The train-test split ratio stands at 50:50.

To ensure the proposed framework is scalable to outdoor environments where the vehicle images are affected by external disturbances such as light illumination, translation, scaling, etc., another experiment is run on the Surveillance-Nature Comprehensive Cars (CompCarsSV) dataset (Yang et al., 2015). This dataset encompasses only the images taken by surveillance cameras and hence performance of the framework can serve as a reference for deployment in outdoor conditions. The training and testing images are 31,148 and 13,333, respectively. Table 5.2 tabulates the statistics of the datasets where μ_{cls} is the mean image count per class and σ_{cls} is the standard deviation of image count per class.

Table 5.2: Datasets Statistics

Dataset	#Train	#Test	#Classes	μ_{cls}	σ_{cls}
CompCarsWeb (Yang et al., 2015)	36,456	15,627	431	84.6	22.1
Stanford Cars (Krause et al., 2013)	8,144	8,014	196	41.6	4.3
Car-FG3K (Wu et al., 2022)	10,676	10,661	1,892	5.6	6.4
CompCarsSV (Yang et al., 2015)	31,148	13,333	281	110.9	84.1

5.4.2 Implementation Details

CGNet uses ResNet50 as its backbone. During training, random cropping is performed before resizing the crop to 224×224. Random horizontal flipping is applied to avoid overfitting. For the optimizer, Stochastic Gradient Descent (SGD) with 0.9 momentum and 5e-4 weight decay is opted. The learning rate is initialized as 0.01 and it decreases by a factor of 10 every 40 epochs. The training is carried out using a batch size of 64 for 90 epochs. Unless stated otherwise, no bounding boxes annotation is used in the training and

testing pipelines and cross entropy loss without label smoothing is used as the loss function.

5.5 Results & Discussions

5.5.1 Quantitative Analysis

Table 5.3 illustrates the existing state-of-the-art works on CompCarsWeb (Yang et al., 2015). On 224×224 resolutions, by setting $N_{CB} = 4$, $d = 3$ and $N_{FC} = 1$, CGNet reports 98.3% accuracy, bringing 1.4% improvement over the baseline. ViT (Dosovitskiy et al., 2020) (86.1M) and TransFG (He et al., 2021) (86.2M) report 96.2% and 96.7% accuracies, respectively upon being fine-tuned based on the steps described in Chapter 5.3.2. CGNet (40.0M) outperforms them by a substantial margin whilst being $2 \times$ lighter in terms of the number of parameters. SWP (Hu et al., 2017) reports 97.6% accuracy but their method is not robust against changes in position and scale as the learned attention masks are static (Ghassemi et al., 2019). Ghassemi et al. (2019) apply STN (Jaderberg et al., 2015) and Wide ResNet50 (Zagoruyko & Komodakis, 2016) as localization and classifier modules, respectively. Although they report 97.8% accuracy, STN is hard to optimize (Hanselmann & Ney, 2020b) and requires a complex training methodology. Recurrent Attention Unit (RAU) (Ma & Boukerche, 2020) and Channel Max Pooling (CMP) (Ma et al., 2019) are trained with bounding box information and they deliver 97.8% and 97.9% accuracies, respectively. Multi-Scale Discriminative Regions Attention Network (MS-DRAN) (Rong et al., 2021) synthesizes multi-granularity features with the help of FPN. It also utilizes the backpropagated gradients in alienating the discriminative regions to suppress the interference of irrelevant information. The network which is trained with cross entropy and interclass ranking loss achieves 98.1% accuracy. Lightweight RAU (LRAU) (Boukerche & Ma, 2021) generates attention states and masks to guide the network in concentrating on the prominent vehicle regions. It presents the same performance as CGNet but it mainly benefits from the bounding box information.

Better performance can be delivered if the downsampling strategy i.e. 1×1 , stride 2 convolution is revised to prevent information loss.

When being trained and evaluated on 448×448 resolutions, CGNet reports 98.6% accuracy, outperforming the Global Topology Constraint Network (GTCNet) (Xiang et al., 2019) which considers inter-part topology relationships by 0.1%.

Overall, the CG module elevates the performance of the backbone network as it incorporates small-scale and large-scale components when modeling the class boundaries. By coalescing the scale-specific representations through the FE and FI stages, it renders a good mix between the local and high-level details which eventually enriches the feature representations.

Table 5.3: Performance Benchmarking on CompCarsWeb

Reference	Backbone	224 ²	448 ²
ResNet50 (He et al., 2016)	-	96.9%	-
ConvAM (Yu et al., 2020)	ResNet50	95.3%	-
A3M (Han et al., 2018)	ResNet50	95.4%	-
Co-occurrence Learning (Elkerdawy et al., 2018)	ResNet50	95.6%	-
Fine-Tuning DARTS (Tanveer et al., 2021)	-	95.9%	-
ViT (Dosovitskiy et al., 2020)	ViT_B_16	96.2%	-
TransFG (He et al., 2021)	ViT_B_16	96.7%	-
SWP (Hu et al., 2017)	ResNet101	97.6%	-
MAWNet (Ghassemi et al., 2019)	Wide ResNet50	97.8%	-
RAU (Ma & Boukerche, 2020)	ResNet101	97.8%	-
CMP (Ma et al., 2019)	DenseNet161	97.9%	-
MS-DRAN (Rong et al., 2021)	ResNet50	98.1%	-
LRAU (Boukerche & Ma, 2021)	ResNet50	98.3%	-
GTCNet (Xiang et al., 2019)	DenseNet264	-	98.5%
CGNet	ResNet50	98.3%	98.6%

On Stanford Cars (Krause et al., 2013), $\mathcal{E} = 0.1$ is adopted to prevent overfitting. N_{CB} , d and N_{FC} are set as 4, 3 and 2, respectively. For fair comparison with the works based on 224×224 resolutions, bounding box information is used. CGNet achieves exceptional

performance by reporting 94.7% accuracy as shown in Table 5.4. The improvement brought by the CG module is 1.1%.

Table 5.4: Performance Benchmarking on Stanford Cars

Reference	Backbone	224 ²	448 ²
ResNet50 (He et al., 2016)	-	93.6%	93.4%
SWP (Hu et al., 2017)	ResNet101	93.1%	-
ConvAM (Yu et al., 2020)	ResNet50	93.1%	-
CA-MSNet (Maopeng Li et al., 2022)	ResNet50	93.5%	-
RAU (Ma & Boukerche, 2020)	ResNet101	93.6%	-
CMP (Ma et al., 2019)	DenseNet161	93.7%	-
LRAU (Boukerche & Ma, 2021)	ResNet50	93.9%	-
MS-DRAN (Rong et al., 2021)	ResNet50	94.3%	-
PCB-MFF (L. Wang et al., 2022)	ResNet34	-	93.4%
PyConv (G. Wang et al., 2021)	ResNet50	-	93.6%
ViT (Dosovitskiy et al., 2020)	ViT_B_16	-	93.7%
FIFNet (P. Wang et al., 2022)	ResNet101	-	94.1%
GTCNet (Xiang et al., 2019)	DenseNet264	-	94.3%
Cross-X Learning (Luo et al., 2019)	ResNet50	-	94.6%
TransIFC+ (H. Liu et al., 2023)	Swin_B	-	94.7%
TransFG (He et al., 2021)	ViT_B_16	-	94.8%
DADAINet (Zhu & Li, 2022)	ResNet50	-	94.9%
MMALNet (F. Zhang et al., 2021)	ResNet50	-	95.0%
PMG (Du et al., 2020)	ResNet50	-	95.1%
EfficientNetv2-L (Tan & Le, 2021)	-	-	95.1%
PSDPNet (Guo et al., 2022)	ResNet50	-	95.1%
APCNN (Ding et al., 2021)	ResNet50	-	95.3%
APINet (Zhuang et al., 2020)	ResNet101	-	95.3%
FCA (Jung et al., 2023)	ResNet50	-	95.3%
CGNet	ResNet50	94.7%	95.4%

For 448×448 resolutions, no bounding box information is used and CGNet tops the rank with 95.4% accuracy. Transformer-based architectures like ViT (Dosovitskiy et al., 2020), TransIFC+ (H. Liu et al., 2023) and TransFG (He et al., 2021) report 93.7%, 94.7% and 94.8% accuracies, respectively. EfficientNetv2-L (Tan & Le, 2021) which is a CNN proposed recently can achieve a balance between training speed and parameter efficiency. Despite having a higher number of parameters (112M), its accuracy is 0.3% lower than CGNet.

Multi-Branch and Multi-Scale Learning Network (MMALNet) (F. Zhang et al., 2021) is a part-based solution that considers feature responses from the upper pyramidal levels in raising the positioning performance of vehicle objects and parts. It delivers 95.0% accuracy but the localization strategy that requires multiple forward passes for a single image is computationally costly. Progressively Sampling Discriminative Parts Network (PSDPNet) (Guo et al., 2022) demonstrates stronger localization power than MMALNet (F. Zhang et al., 2021) where it applies Class Activation Mapping (CAM) (Yu et al., 2020) and spatial correlation matrix for saliency-based feature refinement before extracting the vehicle parts. It reports 95.1% accuracy but the design is counter-intuitive as it requires the number of discriminative parts to be specified beforehand.

Among the attention-based frameworks i.e. GTCNet (Xiang et al., 2019), Data Augmented Dual-Attention Interactive Network (DADAINet) (Zhu & Li, 2022) and APINet (Zhuang et al., 2020), APINet renders performances comparable to CGNet with discrepancy as low as 0.1% accuracy through the exploitation of cross image interaction to uncover the common and unique vehicle features among pairs of images. Nevertheless, it requires a customized image pair construction strategy for optimum compare and contrast process during training. In contradiction, the training pipeline of CGNet is highly generalizable to other datasets.

Compared with the networks from the multi-scale features domain, CGNet outflanks both Attention Pyramid (AP) CNN (Ding et al., 2021) and FCA (Jung et al., 2023) by a narrow 0.1% margin. APCNN is criticized for its reliance on FPN to construct trans-scale features whereas the design of recursive feature aggregation in FCA refrains thorough information propagation from all pyramid levels. The CGNet rectifies these weaknesses by complementing every scale-oriented component with all features from other scales to render more comprehensive and inclusive final feature representations.

Table 5.5 compares the performance of CGNet with other state-of-the-art networks on Car-FG3K (Wu et al., 2022). Since these networks do not report their performances on Car-FG3K in the original works, they are trained according to the details explained in Chapter 5.3.2 and the results are recorded.

Table 5.5: Performance Benchmarking on Car-FG3K

Reference	Backbone	224 ²
ResNet50 (He et al., 2016)	-	84.2%
TransFG (He et al., 2021)	ViT_B_16	78.2%
APCNN (Ding et al., 2021)	ResNet50	82.3%
APINet (Zhuang et al., 2020)	ResNet101	83.5%
ConvNeXt-S (Zhuang Liu et al., 2022)	-	84.1%
SwinV2-S (Ze Liu et al., 2022)	-	84.5%
MMALNet (F. Zhang et al., 2021)	ResNet50	84.9%
LRAU (Boukerche & Ma, 2021)	ResNet50	84.9%
Cross-X Learning (Luo et al., 2019)	ResNet50	85.0%
PMG (Du et al., 2020)	ResNet50	85.3%
CGNet	ResNet50	86.4%

Employing the same configuration as CompCarsWeb (Yang et al., 2015) for CGNet, there is a 2.2% astounding performance leap from the baseline network, recording 86.4% accuracy. Despite rendering the same performance level in Stanford Cars (Krause et al., 2013), APINet (Krause et al., 2013) outflanks APCNN (Ding et al., 2021) on Car-FG3K with 1.2% accuracy and both of them are superior to TransFG (He et al., 2021). ConvNeXt-S (Zhuang Liu et al., 2022), which delivers 94.1% accuracy, is an improvised version of ResNet50 in terms of classification performance and parameter efficiency due to the alteration of macro and micro designs, utilization of grouped convolution, and inverted bottleneck as well as large convolution kernel size. Since the network is studied for general image classification tasks, it is believed that inserting a module that is designed for VMMR will enhance the performance further. Shifted Windows (Swin) Transformer (Ze Liu et al., 2022) addresses the quadratic complexity of the Multi-Head Self-Attention (MHSA) by deploying a shifted windowing scheme in computing spatial

relevancy. Although the local window-based MHSA has linear complexity, it has incapacitated the ability to model global interactions (Qin et al., 2022). The performance of SwinV2-S is 1.9% lower than that of CGNet. Cross-X Learning (Luo et al., 2019) performs slightly better than MMALNet (F. Zhang et al., 2021) and LRAU (Boukerche & Ma, 2021). It attains 85.0% accuracy by leveraging One-Squeeze Multi-Excitation to synthesize multi-scale features. PMG (Du et al., 2020) lags CGNet by 1.1%. By comparing Tables 5.4 and 5.5, the performance difference between PMG and CGNet is aggravated when the number of classes increases. This has proven the superior cross-granularity feature synthesis ability of the CG module, especially when the number of training images per class is scarce.

Table 5.6 tabulates the performance of the state-of-the-art frameworks on CompCarsSV (Yang et al., 2015). Using $N_{CB} = 4, d = 2$ and $N_{FC} = 2$, CGNet delivers 99.1% accuracy. It performs better than CNNs such as AlexNet, Overfeat and GoogLeNet (Yang et al., 2015). In part-based domain, Bilgari *et al.*'s latent SVM and cascade classifier (Biglari et al., 2017a) has a long inference time while Fang *et al.*'s hard attention cropping (Fang et al., 2016) has minimal tolerance against affine transformation. The mid-level features produced by LLC (Wang et al., 2010) are not holistic enough as it reports 98.4% accuracy (Nazemi et al., 2020). Lightweight (LW) CNN (Q. Zhang et al., 2018) adopts a combined learning strategy to perform knowledge distillation but the experiment has seen performance degradation from its backbone network. RAU (Ma & Boukerche, 2020) reports 98.8% accuracy and the performance is also slightly lower than CGNet. Feature Fusion Car Model Classification Network (FF-CMNet) (Yu et al., 2018) learns visual representations of the upper and lower parts independently and the learned features are integrated via FusionNet. Their work reports 98.9% accuracy. For Multi-Agent Systems (MAS) (Amirkhani & Barshooi, 2022), an ensemble of CNNs that fits on the vehicle parts segmented by You Only Learn One Representation (YOLOR) (C.-Y. Wang et al., 2021)

is presented. It claims 98.9% accuracy but the vehicle front image is strictly required to ensure the availability of all targeted regions of interest, including the headlight, upper grill, fog light and bumper. A novel network architecture deduced via the fine-tuning Differential Architecture Search (DARTS) methodology delivers 99.2% (Tanveer et al., 2021). The robustness of the network requires further validation since it has not been inspected under the multi-view scenarios.

Table 5.6: Performance Benchmarking on CompCarsSV

Reference	Backbone	224 ²
Latent SVM and Cascade Classifier (Biglari et al., 2017a)	-	97.5%
AlexNet (Yang et al., 2015)	-	98.0%
Overfeat (Yang et al., 2015)	-	98.3%
GoogLeNet (Yang et al., 2015)	-	98.4%
Dense-SIFT+BoF+LLC (Nazemi et al., 2020)	-	98.4%
Coarse-to-Fine CNN (Fang et al., 2016)	CaffeNet	98.6%
LW CNN with Combined Learning Strategy (Q. Zhang et al., 2018)	VGG16	98.7%
RAU (Ma & Boukerche, 2020)	ResNet50	98.8%
FF-CMNet (Yu et al., 2018)	-	98.9%
MAS (Amirkhani & Barshooi, 2022)	Ensemble of CNNs	98.9%
Fine-Tuning DARTS (Tanveer et al., 2021)	-	99.2%
CGNet	ResNet50	99.1%

5.5.2 Qualitative Analysis

To visually inspect the efficacy of CGNet as compared to the baseline, Gradient-Weighted-CAM (Grad-CAM) (Selvaraju et al., 2017) is applied on the final convolution layer. It is chosen as the visualization technique as it is highly class-discriminative, requires no modification of underlying CNN architectures and is able to produce the visualization in one shot as opposed to Guided Backpropagation (Springenberg et al., 2014), Deconvolution (Zeiler & Fergus, 2014), Class Activation Mapping (Yu et al., 2020), Oquab et al. (2014)'s work and Marginal Winning Probability (J. Zhang et al., 2018).



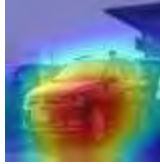
















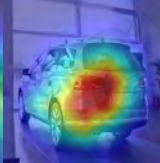
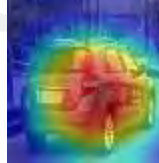



Specifically, to apply Grad-CAM, $I_{CB,A}$ and I_{FI} are selected as the target layer for baseline (ResNet50) and CGNet, respectively. Grad-CAM dissects the convolutional feature maps with respect to the target class and eventually produces a localization map by observing the flow of gradient information. The generated localization map provides an avenue for the inspection of the image regions attended by the network and lends insights into the cause of underperformance.

Table 5.7 illustrates the activation maps overlaid on the 4 randomly selected vehicle models from CompCarsWeb (Yang et al., 2015). For each vehicle model, a few viewpoints are presented. The red regions symbolize class-specific discriminative regions whereas the blue regions are less discriminative.

It is observed that the focused area of the baseline is non-specific. It covers most of the vehicle body which is highly general across different vehicle models. It is also worth noting that the model even dedicates part of the attention to the background which contains noise and it is detrimental to the classification task.

On the contrary, the focused area of CGNet is more concise than that of the baseline. The focused area covers the vehicle parts that are discriminative for VMMR. CGNet localizes the granular and essential vehicle parts like the vehicle logo, headlight, fog light and backlight. Other vehicle parts such as bumpers, side mirrors, hoods, etc. are dropped as they do not present stark differences across different vehicle models.

Table 5.7: Visualization of Feature Maps Using Grad-CAM

Lexus RX Hybrid		Volkswagen Variant		Dongfeng Fengshen H30		MG6	
Baseline	CGNet	Baseline	CGNet	Baseline	CGNet	Baseline	CGNet
							
							
							

5.5.3 Ablation Study

An ablation study is conducted with 224×224 resolutions to investigate how each hyperparameter in the CG module affects the performance of CGNet. The figures reported in Table 5.8 are computed based on the CompCarsWeb (Yang et al., 2015). As the datasets are imbalanced, precision, recall and f1-score are also reported for holistic evaluation. The number of parameters and floating point operations (FLOPs) are attached as an indication of the computational complexity.

Table 5.8: Ablation Study on CompCarsWeb

Exp.	N_{CB}	d	N_{FC}	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	#Params (M)	GFLOPs
Baseline	-	-	-	96.9	97.0	96.6	96.7	24.4	4.1
A	2	-	2	97.9	98.0	97.7	97.8	52.3	7.0
	3	2	2	98.1	98.2	97.9	98.0	44.4	9.1
	4	2	2	98.1	98.2	98.0	98.1	40.4	15.0
B	4	1	2	98.1	98.1	97.8	97.9	40.4	15.0
	4	2	2	98.1	98.2	98.0	98.1	40.4	15.0
	4	3	2	98.2	98.3	98.1	98.1	40.4	15.0
	4	4	2	98.0	98.1	97.8	97.9	40.4	15.0
C	4	2	1	98.2	98.3	98.0	98.1	40.0	15.0
	4	2	2	98.1	98.2	98.0	98.1	40.4	15.0
	4	3	1	98.3	98.4	98.1	98.2	40.0	15.0
	4	3	2	98.2	98.3	98.1	98.1	40.4	15.0

Experiment A aims to investigate the number of CBs that should be channeled into the CG module for the learning of multi-granularity features. In particular, the effect of $N_{CB} \in \{2, 3, 4\}$ is determined empirically. All the metrics show consistent observation. That is the model performance improves when $I_{CB,i}$ from early layers are incorporated into the CG module. This suggests that low-level feature maps which are rich in structural information are pivotal and they complement the abstract and high-level details in high-level feature maps. For $N_{CB} = 4$, 98.1% accuracy is reported.

As low-level feature maps are large in resolution, applying dilated convolution can effectively expand the receptive fields without increasing the number of trainable

parameters. Experiment B is carried out to realize this where different values of $d \in \{1, 2, 3, 4\}$ are applied for 3×3 convolution in the FE stage of the CG module. Setting $d = 1$ signifies normal convolution where the receptive fields are not expanded. For $d > 1$, the receptive fields are broadened to facilitate the capture of contextual information that is not in close proximity. The result suggests that increasing d brings a positive impact on model performance and 98.2% accuracy is achieved when $d = 3$. The model performance begins to drop when $d = 4$. This is because the CG module fails to aggregate the key tiny discriminative local features due to spatial inconsistency (Hamaguchi et al., 2018).

Experiment C is carried out to compare the model performance when one or two fully connected layers is employed as classification head. Aside from N_{FC} , another varying element in Experiment C is $d \in \{2, 3\}$. The result shows that the CG module is capable of capturing both local and global discriminative features to produce rich representations and hence using two fully connected layers to learn additional features is unnecessary. By setting $d = 3$ and $N_{FC} = 1$, CGNet achieves 98.3% accuracy.

In terms of computational cost, increasing N_{CB} lowers the parameter counts but raises the FLOPs exponentially. Sweeping N_{CB} from 2 to 4, the network experiences parameter reduction of 8M and 4M whereas the increment in FLOPs amounts to 2 GFLOPs and 6 GFLOPs. This is because a higher N_{CB} results in lower C_{FE} for the FE stage which in turn slashes the number of filters to learn for the layers with high channel counts considerably. For FLOPs, the increment is due to the large resolution feature maps from the early convolution layers which leads to more convolution operations being executed. Varying d and the N_{FC} induce negligible change with the network wandering around 40.0M parameters and 15.0 GFLOPs.

To summarize, N_{CB} is the most influential parameter that alters the classification performance and computational costs of CGNet. Under the optimum setting ($N_{CB} = 4, d = 3, N_{FC} = 1$), CGNet renders 98.3% accuracy with 40.0M parameters and 15.0 GFLOPs.

5.5.4 Generalization Study

Figure 5.3 shows the results of the experiment where the compatibility between the CG module and existing CNNs is examined on CompCarsWeb (Yang et al., 2015). Incorporating the CG module into ResNet50 (He et al., 2016) demonstrates the largest leap in performance, which is 1.4%. The performance of DenseNet169-based (Huang et al., 2017) CGNet is also remarkable where it achieves 98.2% accuracy, an improvement of 0.9% over the baseline. Inceptionv3 (Szegedy et al., 2016) uses multi-size convolution kernels to capture multi-scale information via the Inception module. Despite the similar motivation between Inception and CG module, combining CG module with Inceptionv3 still brings 0.4% improvement.

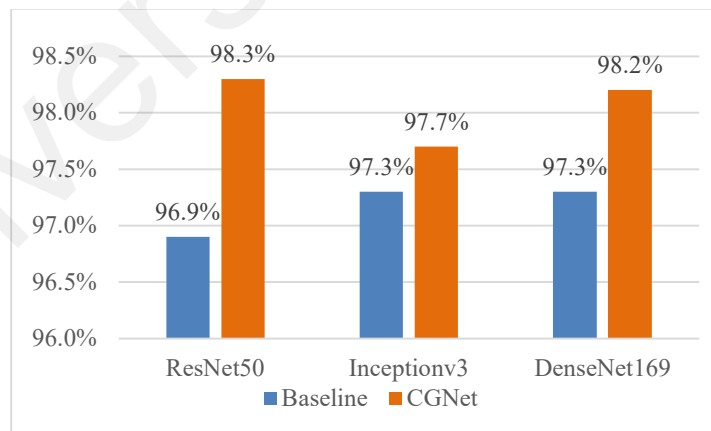


Figure 5.3: Compatibility between CG Module and Existing CNNs on CompCarsWeb

5.6 Conclusion

In this chapter, a CG module that is responsible for multi-scale feature extraction and integration is proposed. Although high-level feature maps carry strong semantic

information, using them alone for VMMR limits the model performance as the fine-grained details from low-level feature maps which become the key differentiator between various vehicle models are forgone. This issue is addressed by introducing the CG module to leverage the feature maps generated from various pyramid levels. The resultant feature maps produced by the CG module are more extensive as they encompass exquisite low-level attributes and dense high-level details. This is validated through the experiments where it brings a 1.4% improvement on the CompCarsWeb dataset. Remarkable performance is also seen in Stanford Cars, Car-FG3K and CompCarsSV datasets. Furthermore, the qualitative analysis reveals that the CG module pays attention to the discriminative details and its focus area is more concise as compared to the baseline. The baseline without multi-scale information fails to alienate the common vehicle parts from its attention and this reduces the classification performance. The choice of hyperparameters is justified in the ablation study and the generalization study suggests that the CG module is highly compatible with other CNNs.

CHAPTER 6: COARSE-TO-FINE CONTEXT AGGREGATION NETWORK FOR VEHICLE MAKE AND MODEL RECOGNITION

6.1 Introduction

This chapter continues the study in the multi-scale features domain which aims to distill both semantic information from large-scale components and detailed information from small-scale components to enrich the deep feature embeddings. A meticulously designed Coarse-to-Fine Context Aggregation (CFCA) module that is more parameter-efficient is deliberated. The advantages of the CFCA module come in three aspects. Firstly, at each pyramidal level, it performs multiple dilated convolutions on the input feature maps to expand the field of view of convolution kernels so that more granular and crucial details of the vehicles can be extracted. Secondly, it minimizes computational complexity by employing grouped convolution for feature extraction without sacrificing the network performance. Thirdly, through 1×1 convolution, the scale-varying components are aggregated adaptively based on importance to form exclusive multi-scale features. The contributions of this chapter are summarized as follows:

- Present CFCA module to generate scale-aware feature embeddings that encapsulate shallow-level fine-grained details and deep-level semantic information by aggregating the features from various convolution layers
- Conduct performance benchmarking and demonstrate the high compatibility between the CFCA module and other backbone Convolutional Neural Networks (CNNs)
- Visualize the feature maps refined by the CFCA module to illustrate the ability to capture discriminative components of the vehicle upon considering coarse-to-fine features

6.2 Literature Review

Feature-based methods are frequently adopted during the pre-deep learning era. They utilize the raw features such as gradient orientation and interest points information to characterize the vehicle attributes. The well-known feature extractors include Histogram of Oriented Gradient (HOG), Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF). Manzoor et al. (2019) opted for HOG and presented satisfactory results for 35 vehicle models. However, the handcrafted features are susceptible to external disturbances and this deteriorates the classification performance in real-life scenarios (Chan et al., 2015; Dong et al., 2015; Ge et al., 2017; Huang et al., 2015; S. Li et al., 2018). Hence, several feature encoding techniques are used to perform feature transformation to improve the robustness. Jamil et al. (2020) introduced Bag of Expressions (BoE) to mitigate the effect of viewpoint variation whereas Nazemi et al. (2020) adopted Locality-constraint Linear Coding (LLC) to encode dense SIFT features. Despite the efficacy of these feature encoding schemes, the classification performance is heavily dependent on the choice of feature extractor algorithms that requires an experience-based decision.

Since the background scene is irrelevant to the vehicle recognition task and degrades the classification performance, part-based methods seek ways to isolate the vehicle object from the scene. Multiscale Attention Windows Network (MAWNet) (Ghassemi et al., 2019) proposes the learning of localization and classification tasks in a unified framework using Spatial Transformer Network (STN) (Jaderberg et al., 2015) and Wide ResNet50 (Zagoruyko & Komodakis, 2016), respectively. However, the bounding box annotation is required in the pre-training stage. Another branch of work under part-based methods learns localization tasks in a weakly supervised way where they rely on image labels only. Multilayer Bilinear Pooling Network (MLBPN) presented by Ming Li et al. (2022) segments the vehicle by binarizing the activation responses and enhances the feature

representations through bilinear pooling. The huge embeddings produced by bilinear pooling are less practical (X. Liu et al., 2022) due to high computational complexity. Progressively Sampling Discriminative Parts Network (PSDPNet) (Guo et al., 2022) similarly localizes the vehicle by slicing the region that exhibits peak responses. The prominent vehicle parts are subsequently identified through a spatial correlation matrix. PSDPNet requires single-view images to achieve consistency in sampling the discriminative parts.

Instead of removing the seemingly non-discriminative parts completely, Yu et al. (2020), Boukerche and Ma (2021) and Elkerdawy et al. (2018) applied soft attention masks to enhance the prominent vehicle region and suppress the noise. Convolutional Attention Model (ConvAM) (Yu et al., 2020) features two Long Short-Term Memory (LSTM)-based attention mechanisms to characterize inter-channel and inter-spatial relationships. The network size is huge due to Convolutional LSTM operations on 2048-channel feature maps. Elkerdawy et al. (2018) presented co-occurrence learning to exploit spatial dependency. To reduce the computation cost, the co-occurrence layer performs channel reduction but this compromises the classification performance (Forcen et al., 2020). Lightweight Recurrent Attention Unit (LRAU) (Boukerche & Ma, 2021) is a modular attention module that successively refines the attention state based on the input feature maps. Although it successfully highlights the discriminative regions, utilization of stride 2, 1×1 convolution to downsize feature maps causes information loss

There is also another line of work focusing on increasing the feature expressive ability of the backbone networks by proposing novel CNNs and transformer architectures as evident in the works of Dosovitskiy et al. (2020) and Tan and Le (2021). Tan and Le (2021) revised EfficientNet (Tan & Le, 2019) into EfficientNetv2 built from MBConv and Fused-MBConv. They also put forward a progressive learning strategy in which the

regularization strength is adjusted according to the image resolution to achieve faster training speed, better parameter efficiency and higher accuracy. Since EfficientNetv2 is designed for general image classification tasks, it is believed that the classification performance of EfficientNetv2 can be boosted further in the fine-grained vehicle classification domain if customization is done. Motivated by the astounding performance of transformer architecture in the Natural Language Processing (NLP) domain, Dosovitskiy et al. (2020) proposed a Vision Transformer (ViT) for Computer Vision (CV) applications. ViT treats the image as a sequence of image patches and it generates deep feature embeddings through encoder layers. Nevertheless, the failure of class token in summarizing information from all image tokens causes it to be less competitive (Kang et al., 2022; Touvron et al., 2021; Yuan et al., 2021).

One of the past studies in the multi-scale features domain is by L. Wang et al. (2022). Multi-stream convolutions are implemented through Parallel Convolutional Block (PCB) which comprises 1×1 and 3×3 convolutions. The multi-layer features are then consolidated through matrix outer product and normalization. G. Wang et al. (2021) introduced Pyramid Convolution (PyConv) as the building block of ResNet50 (He et al., 2016) where convolution kernels of various sizes are used to learn multi-scale information. P. Wang et al. (2022) proposed Feature Integration and Feature Fusion Network (FIFNet) where a simple summation is performed between feature maps from the last two layers to retain the global and subtle information. It is important to note that the final feature embeddings of L. Wang et al. (2022), P. Wang et al. (2022) and G. Wang et al. (2021) still lack tiny vehicle cues since only feature maps from the top pyramid level are used. The CFCA module overcomes this by establishing lateral connections from multiple pyramid levels of the backbone network to achieve thorough information transmission in modeling coarse-to-fine information.

Cross-Layer Attention Network (CLANet) (Huang et al., 2022) contains Cross-Layer Context Attention (CLCA) and Cross-Layer Spatial Attention (CLSA) to mix the local texture information and global context features and recalibrate features based on spatial importance, respectively. Their proposal incurs a large memory footprint since the computation involves feature maps of large spatial resolutions. On the contrary, the CFCA module limits the memory footprint by synthesizing the cross-layer information using small-resolution feature maps within the adjacent scales. Progressive Multi-Granularity (PMG) (Du et al., 2020) learns scale-aware features by appending a classifier at each scale level. This strategy leads to suboptimal performance since the shallow and deep layer classifiers require the shallow layer to learn both semantic and spatial information. In the CFCA module, a single classification head is added after the last CFCA module to enable every layer to focus on the learning of scale-specific features. Feature Pyramid Network (FPN) (Lin et al., 2017a) is a well-known technique and it has laid the foundation for several works in the multi-scale features domain (S. Liu et al., 2018; Luo et al., 2019; Peng et al., 2021). The CFCA module takes inspiration from FPN but the simple feature transfer method between layers is revamped to effectively collect and integrate scale-varying components in stages.

6.3 Methodology

In fine-grained vehicle recognition tasks, relying on top-level feature maps alone hinders the full potential of CNN in learning the subtle features. As the CNN grows deeper, the size of the feature maps diminishes and the loss of microscopic features is inevitable. However, the pyramidal features produced by CNN carry various degrees of semantical information and all of them contribute to identifying the subtle details of different vehicle classes. The low-level feature maps produced by early convolutional blocks (CBs) pay attention to the geometric details i.e. colour, edges and corners but they are semantically weak. The high-level feature maps generated by CBs at deep layers

contain dense abstract features but they lack spatial details. Since none of them are impeccable and they complement each other, a CFCA module is presented to integrate these hierarchical features to elevate the representation ability of the network.

6.3.1 Coarse-to-Fine Context Aggregation Module

Figure 6.1 portrays the structure of the CFCA module. The CFCA module is a unit that takes coarse-grained and fine-grained features produced by CNN and recursively processes them. It adds a bottom-up pathway that is parallel to the CBs. The objective is to recover the small-scale features that diminish with the convolution operations as well as retain the large-scale features by consolidating semantic and detailed information from all levels of feature maps to construct scale-aware feature maps.

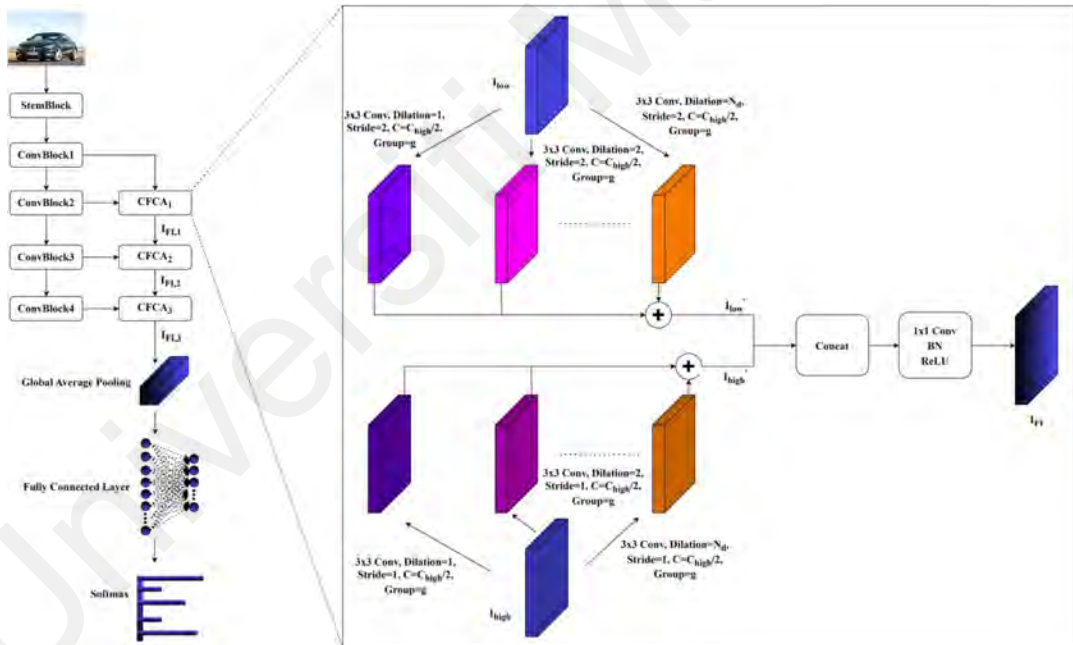


Figure 6.1: Coarse-to-Fine Context Aggregation Network

Given an input image $I_i \in \mathbb{R}^{H \times W \times C}$ where H, W and C are height, width and number of channels, it undergoes a series of convolutions in CNN to produce feature maps of different spatial sizes. A CNN usually consists of a stem unit to drastically downsize the resolution of I_i followed by CB for feature extraction purposes. The stem unit produces

$I_{stem} \in \mathbb{R}^{H_{stem} \times W_{stem} \times C_{stem}}$ whereas the CBs produce low-level, mid-level and high-level feature maps. These feature maps are denoted as $\{I_{CB,i}\}_{i=1}^{N_{CB}} \in \mathbb{R}^{H_{CB,i} \times W_{CB,i} \times C_{CB,i}}$ where N_{CB} is the number of convolutional blocks and it is equal to 4 for most networks.

Let $\{CFCA_i\}_{i=1}^{N_{CFCA}}$ be the i^{th} out of N_{CFCA} modules, it receives two inputs at a time, namely $I_{low,i} \in \mathbb{R}^{H_{low,i} \times W_{low,i} \times C_{low,i}}$ and $I_{high,i} \in \mathbb{R}^{H_{high,i} \times W_{high,i} \times C_{high,i}}$, which are the low-level and high-level feature maps. It is important to note that $I_{low,1}$ is the feature maps generated by backbone CNN whereas $\{I_{low,i}\}_{i=2}^{N_{CFCA}}$ is the feature maps generated by $CFCA_{i-1}$. The operations within the CFCA module can be broken down into two, namely the Feature Extraction (FE) and Feature Integration (FI) stages. During the FE stage, convolution operations are performed to extract more discriminative features. It is worth noting that due to the limited size of the receptive field, a typical CNN cannot track long-range dependencies and this inhibits the information exchange between spatial locations that are far apart from each other. To alleviate the weakness, instead of a normal convolution, $k \times k$ dilated convolution is opted to enlarge the receptive field while keeping the number of trainable parameters intact. In particular, the area of the receptive field is configurable by setting the dilation rate $d \in \{1, 2, \dots, N_d\}$ where N_d is the maximum dilation rate in use. Since the vehicle sizes vary across the images, there is hardly a single dilation rate that suits all cases. The issue of scale variations is addressed by performing multi-branch convolutions where every branch performs convolution with different d in parallel and the resultant feature maps are subsequently consolidated via summation operation. In designing the CFCA module, the model complexity is taken care of by employing grouped convolution to keep the number of parameters minimal. In addition, special precautions are taken to ensure the spatial size of resultant feature maps $I_{low,i}'$ is same as $I_{high,i}'$ for feature map integration in the FI stage. To conform to this

requirement, strided convolution is performed where stride s is set appropriately depending on the backbone networks. The operation performed on $I_{low,i}$ is represented as

$$I_{low,i}' = \sum_{d=1}^{N_d} Conv_{3 \times 3, \frac{C_{high,i}}{2}, d, 2, g}(I_{low,i}) \quad (6.1)$$

The convolution operation can be generalized as $Conv_{k \times k, C_{out}, d, s, g}$ where C_{out} is the number of output channels and g is the number of groups for grouped convolution.

When processing $I_{high,i}$, the same design principle is adopted where dilated grouped convolution is utilized to deduce the features with minimal increase in the number of parameters. Convolution with $s = 1$ is used to process $I_{high,i}$ during the FE stage. The operation performed on $I_{high,i}$ is represented as

$$I_{high,i}' = \sum_{d=1}^{N_d} Conv_{3 \times 3, \frac{C_{high,i}}{2}, d, 1, g}(I_{high,i}) \quad (6.2)$$

Dilated convolution has widened the field of view of the convolution filters but the learned features are still scale-specific. It is conjectured that feature maps with different granularities should be aggregated together to allow the exchange of cross-scale information so that the resultant feature maps have a good mix of local and global features. This consideration is incorporated into the FI stage. In the design, an appropriate mix between $I_{low,i}'$ and $I_{high,i}'$ is learned adaptively through convolution.

$$I_{FI,i} = h([I_{low,i}', I_{high,i}']) \quad (6.3)$$

where $[\bullet]$ is concatenation along the channel axis and h consists of $Conv_{1 \times 1, C_{high}, 1, 1, 1}$, batch normalization (BN) and rectified linear unit (ReLU). $I_{FI,i}$ is the output of $CFCA_i$

module and it is interpreted as the weighted combination between different levels of feature maps with the important feature channel acquiring a larger weight.

After processing the pyramidal features iteratively, the final feature maps, which are $I_{FI,N_{CFCA}}$, carry the holistic cross-granularity information. $I_{FI,N_{CFCA}}$ is then pooled using global average pooling (*GAP*) and is fed into the fully connected layer (*FC*) to generate classification logits.

$$Logits = FC_L \left(GAP(I_{FI,N_{CFCA}}) \right) \quad (6.4)$$

where FC_L is *FC* with L output neurons.

6.3.2 Coarse-to-Fine Context Aggregation Network

The CFCA module is designed to improve the feature learning of existing CNNs. To demonstrate its ability to improve the learning of discriminative and diverse features, four popular CNNs, namely VGG16 (Simonyan & Zisserman, 2014), Inceptionv3 (Szegedy et al., 2016), ResNet50 (He et al., 2016) and DenseNet169 (Huang et al., 2017), are chosen. Generally, these networks are denoted as CFCA Network (CFCANet) upon being integrated with the CFCA module. It is worth noting that the CFCA module is added to higher layers of the CNN first. For instance, when $N_{CFCA} = 3$ and $N_{CB} = 4$, the CFCA module would be added after CB_2, CB_3, CB_4 .

The motivation of VGG16 (Simonyan & Zisserman, 2014) is to reduce the number of trainable parameters by discarding the large convolution kernels. It uses multiple 3×3 convolution kernels to enlarge the receptive field and reduces the training time by a significant margin. During the experiment with VGG16, VGG16 is made even lighter by removing the computationally expensive three-layer FC. A *GAP* layer is added after the top-level feature maps to reduce the feature vector to $1 \times 1 \times 512$. A classification head then

follows to produce the classification logits. To construct VGG16-CFCA, the feature maps after the first maximum pooling layer are regarded as I_{stem} and the output of all subsequent maximum pooling layers as $I_{CB,i}$. In particular, based on $224 \times 224 \times 3$ I_i , the sizes of I_{stem} and $\{I_{CB,i}\}_{i=1}^4$ are $128 \times 128 \times 64$, $56 \times 56 \times 128$, $28 \times 28 \times 256$, $14 \times 14 \times 512$ and $7 \times 7 \times 512$.

Inceptionv3 (Szegedy et al., 2016) is an architecture ameliorated based on GoogLeNet (Szegedy et al., 2015). Keeping computational efficiency in mind, large convolution kernels in the Inception module are factorized into a sequence of 3×3 convolution kernels. To achieve higher computational cost savings, asymmetric convolution is adopted where $k \times k$ convolution is segregated into $1 \times k$ and $k \times 1$ convolution in the intermediate layers. The Inception module also contains convolution and pooling branches for efficient grid size reduction and elimination of representational bottleneck. Based on $299 \times 299 \times 3$ I_i , the feature maps $73 \times 73 \times 64$, $71 \times 71 \times 192$, $35 \times 35 \times 288$, $17 \times 17 \times 768$ and $8 \times 8 \times 2048$ are identified as I_{stem} and $\{I_{CB,i}\}_{i=1}^4$.

ResNet50 (He et al., 2016) resolves the vanishing gradient problem for deep CNN through residual learning. The identity shortcut connection adds no computational complexity and allows the network to achieve a gain in accuracy by growing the network deeper. Based on $224 \times 224 \times 3$ I_i , sizes of I_{stem} is $56 \times 56 \times 64$ and $\{I_{CB,i}\}_{i=1}^4$ are the feature maps produced by every bottleneck building block.

DenseNet169 (Huang et al., 2017) employs dense connections between the layers where each layer accepts the feature maps from all preceding and current layers as input. The dense connection brings several advantages including the construction of a substantially deeper network without being plagued by vanishing gradient problems and stronger feature propagation between layers. Based on $224 \times 224 \times 3$ I_i , the size of I_{stem} is

$56 \times 56 \times 64$ and $\{I_{CB,i}\}_{i=1}^4$ are the feature maps produced by every dense block and transition layer.

6.3.3 Loss Function

The well-known cross entropy loss is employed as a loss function during training to minimize the difference between actual and predicted labels. To optimize the trainable parameters, the cross entropy loss is backpropagated during training and the parameters in CNN are updated to achieve minimal loss. In addition, a uniformly distributed noise is purposely injected into the ground truth labels during the training. Such a regularization technique is called label smoothing (Szegedy et al., 2016) and it is improvised from the cross entropy loss. On one hand, label smoothing loss addresses the overfitting issue by preventing the model from being overconfident with the prediction. On the other hand, it is effective in attenuating the effect of wrongly labeled data which is ineluctable, especially with large-scale datasets. Cross entropy with label smoothing is given as

$$Loss = -\frac{1}{N_{Train}} \sum_{i=1}^{N_{Train}} \sum_{l=1}^L (1 - \mathcal{E}) y_{i,l} \log p_{i,l} + \frac{\mathcal{E}}{(L - 1)} y_{i,l} \log p_{i,l} \quad (6.5)$$

where \mathcal{E} is the label smoothing coefficient, $y_{i,l}$ is one-hot encoded ground truth, $p_{i,l}$ is the output distribution predicted by the model and N_{Train} is the number of training samples.

6.4 Experiments

6.4.1 Datasets

The proposed CFCA module is validated on 5 public datasets. These datasets are either web-crawled images or taken by surveillance cameras. They present different complexities in terms of viewpoint variations, number of classes and number of images. Hence, the obtained classification results can serve as an invaluable reference to assess the performance of the network.

CompCarsWeb (Yang et al., 2015) is the Web-Nature Comprehensive Cars (CompCars) dataset that consists of 431 vehicle models. It contains 36,456 and 15,627 training and testing images, respectively. The crawled images contain various viewpoints including front, rear, side, front-side and rear-side views.

Similar to CompCarsWeb, Stanford Cars (Krause et al., 2013) is a web-nature vehicle recognition dataset. It is a popular public dataset used for performance benchmarking in fine-grained classification tasks. Based on the 50:50 train-test split, there are 8,144 and 8,044 training and testing images, respectively and the images also contain various viewpoints.

Car-FG3K (Wu et al., 2022) is a highly challenging dataset published recently. It contains 1,892 classes but the number of training images is limited. The mean image count for a class is 5.6, which is significantly lower than that of CompCarsWeb (84.6) and Stanford Cars (41.6). The dataset also covers viewpoint variations and no bounding box annotation is provided to remove the background information. The train-test split ratio stands at 50:50.

Since the web-nature images are high in quality and do not reflect the real-life scenario completely, the proposed network is also validated on CompCarsSV (Yang et al., 2015), the Surveillance-Nature CompCars dataset. The images are captured by surveillance cameras and the viewpoint is limited to the vehicle frontal face only. It consists of 31,148 and 13,333 training and testing images, covering both daytime and nighttime images.

A smaller-scale dataset collected by Ali et al. (2022) is used in this chapter. Special precautions are taken to ensure the images collected using surveillance cameras contain real-life circumstances such as occlusion, viewpoint difference and illumination variations. The training and testing image counts are 3,096 and 751, respectively.

Figure 6.2 depicts the sample images and the detailed information about each dataset is recorded in Table 6.1.



Figure 6.2: Sample Images from CompCarsWeb, Stanford Cars, Car-FG3K, CompCarsSV and Mohsin-VMMR

Table 6.1: Datasets Statistics

	CompCarsWeb	Stanford Cars	Car-FG3K	CompCarsSV	Mohsin-VMMR
#Train	36,456	8,144	10,676	31,148	3,096
#Test	15,627	8,014	10,661	13,333	751
#Classes	431	196	1,892	281	48
Nature	Web	Web	Web	Surveillance	Surveillance
Viewpoint	Multi	Multi	Multi	Single	Multi

6.4.2 Implementation Details

The proposed network is implemented using the PyTorch framework. The network is loaded with ImageNet (Deng et al., 2009) pretrained weights and the fine-tuning process follows. To prevent overfitting, data augmentation technique is implemented in the

training pipeline and they include random cropping, random horizontal flipping and resizing. The network is optimized using Stochastic Gradient Descent (SGD) with 0.9 momentum and $5e-4$ weight decay. The initial learning rate is set as 0.01. Unless stated otherwise, ResNet50 (He et al., 2016) is used as the base to develop CFCANet. No bounding box information is used in both the training and testing process.

The proposed network is fine-tuned on CompCarsWeb (Yang et al., 2015), Stanford Cars (Krause et al., 2013), and CompCarsSV (Yang et al., 2015) for 90 epochs and the learning rate is decayed by a factor of 10 for every 40 epochs. For Car-FG3K (Wu et al., 2022), CFCANet is trained for 90 epochs too but cosine annealing is utilized instead. Since the scale of Mohsin-VMMR (Ali et al., 2022) is relatively smaller, fine-tuning is performed for 50 epochs and the learning rate is decayed by a factor of 10 for every 20 epochs. CFCANet is trained based on cross entropy loss for CompCarsWeb, CompCarsSV and Mohsin-VMMR. As for Stanford Cars and Car-FG3K, label smoothing loss with $\mathcal{E} = 0.1$ is adopted.

6.5 Results & Discussions

6.5.1 Quantitative Analysis

Table 6.2 depicts the state-of-the-art networks tested on CompCarsWeb (Yang et al., 2015). The proposed network, CFCANet ($N_{CFCA} = 3, g = 32$ and $N_d = 3$), demonstrates exemplary performance against the existing works where it achieves 98.0% accuracy. As compared to the baseline, which is ResNet50 (He et al., 2016), an improvement margin of 1.1% is recorded. Such observation conveys a message that although the top-level feature maps possess an extensive semantic understanding of various vehicle models, this information alone is insufficient as it lacks detailed spatial details. The shortcoming is addressed by CFCANet where the low-level and high-level information are coupled

together to retain both structural and semantic information. As a result, a better recognition rate is seen.

Table 6.2: Performance Benchmarking on CompCarsWeb

Reference	Backbone	Input	BBox	Accuracy
ResNet50 (He et al., 2016)	-	224 ²	×	96.9%
ConvAM (Yu et al., 2020)	ResNet50	224 ²	×	95.3%
A3M (Han et al., 2018)	ResNet50	224 ²	×	95.4%
Co-occurrence Learning (Elkerdawy et al., 2018)	ResNet50	224 ²	×	95.6%
Fine-Tuning DARTS (Tanveer et al., 2021)	-	224 ²	×	95.9%
ViT (Dosovitskiy et al., 2020)	ViT_B_16	224 ²	×	96.2%
TransFG (He et al., 2021)	ViT_B_16	224 ²	×	96.7%
SWP (Hu et al., 2017)	ResNet101	224 ²	×	97.6%
PLFENet (Lu et al., 2022)	ResNet101	448 ²	×	97.7%
MAWNet (Ghassemi et al., 2019)	Wide ResNet50	224 ²	×	97.8%
RAU (Ma & Boukerche, 2020)	ResNet101	224 ²	✓	97.8%
CMP (Ma et al., 2019)	DenseNet161	224 ²	✓	97.9%
LRAU (Boukerche & Ma, 2021)	ResNet50	224 ²	✓	98.3%
CFCANet	ResNet50	224²	×	98.0%

In addition, several recent works that report their classification performance based on CompCarsWeb are presented. Spatially Weighted Pooling (SWP) (Hu et al., 2017) introduces learnable spatial masks to highlight the important spatial positions. The network reports 97.6% accuracy but the learned spatial masks are translation intolerant. Lu et al. (2022) proposed a Part-Level Feature Extraction Network (PLFENet) that uses feature grouping and feature fusion techniques to learn part information. It uses a stronger backbone which is ResNet101 (He et al., 2016) as compared to CGNet and delivers 97.7% accuracy. MAWNet (Ghassemi et al., 2019) reports 97.8% accuracy. The training methodology is complex as STN and Wide ResNet50 are not trainable end-to-end.

Recurrent Attention Unit (RAU) (Ma & Boukerche, 2020) and LRAU (Boukerche & Ma, 2021) are highly similar whereby they refine the feature maps by generating attention masks and attention states. LRAU (98.3%) achieves better performance than RAU (97.8%) and both of them are trained using bounding box information. Furthermore,

Channel Max Pooling (CMP) (Ma et al., 2019) which is built upon DenseNet161 puts forward a novel pooling strategy to retain the salient information while downsizing the feature maps. It achieves 97.9% accuracy with bounding box information as well.

The work is also compared with the transformer architectures i.e. Vision Transformer (ViT) (Dosovitskiy et al., 2020) and Fine-Grained Transformer (TransFG) (He et al., 2021) given their competitive performance in various computer vision tasks recently. It is important to note that they are trained based on the implementation details described in Chapter 6.3.2. ViT mainly consists of Multi-Head Self-Attention (MHSA) and Feed-Forward Network. To pay more attention to the significant vehicle parts and suppress the noisy patches, TransFG incorporates a Part Selection Module (PSM) to identify the most informative image tokens which are then sent to the classification head. The amelioration proposed in TransFG achieves 96.7% accuracy and it elevates the classification performance of ViT by 0.5% on CompCarsWeb. The strategy of TransFG to focus only on significant image tokens leads to overfitting and hence its performance lags behind CFCANet.

Overall, the proposed CFCA module improves the learning of fine-grained vehicle recognition tasks where it enlarges the receptive field of convolution kernels through dilated convolution to allow information sharing between vehicle parts that are not in close vicinity during the FE stage. Subsequently, these scale-specific feature maps are consolidated together in the FI stage to render the final feature representations that are rich in the local spatial context and global abstract information.

The performance of CFCANet ($N_{CFCA} = 3, g = 32$ and $N_d = 4$) is evaluated on Stanford Cars (Krause et al., 2013) using 224×224 and 448×448 resolutions and the results are shown in Table 6.3.

Table 6.3: Performance Benchmarking on Stanford Cars

Reference	Backbone	Input	BBox	Accuracy
ResNet50 (He et al., 2016)	-	224 ²	✓	93.6%
SWP (Hu et al., 2017)	ResNet101	224 ²	✓	93.1%
ConvAM (Yu et al., 2020)	ResNet50	224 ²	✓	93.1%
CA-MSNet (Maopeng Li et al., 2022)	ResNet50	224 ²	✓	93.5%
RAU (Ma & Boukerche, 2020)	ResNet101	224 ²	✓	93.6%
CMP (Ma et al., 2019)	DenseNet161	224 ²	✓	93.7%
LRAU (Boukerche & Ma, 2021)	ResNet50	224 ²	✓	93.9%
CFCANet	ResNet50	224²	✓	94.5%

ResNet50 (He et al., 2016)	-	448 ²	×	93.4%
PCB-MFF Network (L. Wang et al., 2022)	ResNet34	448 ²	×	93.4%
PyConv (G. Wang et al., 2021)	ResNet50	448 ²	×	93.6%
ViT (Dosovitskiy et al., 2020)	ViT_B_16	448 ²	×	93.7%
MLBPNNet (Ming Li et al., 2022)	ResNet34	448 ²	×	93.8%
EfficientNetv2-S (Tan & Le, 2021)	-	448 ²	×	93.8%
FIFNet (P. Wang et al., 2022)	ResNet101	448 ²	×	94.1%
PLFNet (Lu et al., 2022)	ResNet101	448 ²	×	94.1%
CLANet (Huang et al., 2022)	ResNet101	448 ²	×	94.5%
Cross-X Learning (Luo et al., 2019)	ResNet50	448 ²	×	94.6%
EfficientNetv2-M (Tan & Le, 2021)	-	448 ²	×	94.6%
SA-MFNet (H. Chen et al., 2022)	-	448 ²	×	94.7%
TransIFC+ (H. Liu et al., 2023)	Swin-B	448	×	94.7%
TransFG (He et al., 2021)	ViT_B_16	448 ²	×	94.8%
DADAINet (Zhu & Li, 2022)	ResNet50	448 ²	×	94.9%
MMALNet (F. Zhang et al., 2021)	ResNet50	448 ²	×	95.0%
PMG (Du et al., 2020)	ResNet50	448 ²	×	95.1%
EfficientNetv2-L (Tan & Le, 2021)	-	448 ²	×	95.1%
PSDPNet (Guo et al., 2022)	ResNet50	448 ²	×	95.1%
CFCANet	ResNet50	448²	×	95.1%

For a fair comparison with the existing works, bounding box information is used in the training when the image resolution is 224×224. It is shown that CFCANet brings 0.9% improvement over ResNet50 (He et al., 2016) and reports 94.5% accuracy. Multi-Scale Sparse Network with Cross-Attention mechanism (CA-MSNet) (Maopeng Li et al., 2022) delivers 93.5% accuracy where it features a cross-attention mechanism module and a multi-scale sparse structure module as the building block of CNN to improve the feature expressive ability. As compared to RAU (Ma & Boukerche, 2020), CMP (Ma et al., 2019) and LRAU (Boukerche & Ma, 2021), CFCANet also outperforms them by an average of 0.8% accuracy.

For 448×448 resolution, no bounding box information is utilized. The proposed network ($N_{CFCA} = 3, g = 32$ and $N_d = 4$) reports 95.1% accuracy as compared to ResNet50's 93.4% accuracy. Self-supervised Attention Filtering and Multi-Scale Features Network (SA-MFNet) (H. Chen et al., 2022) and Invariant cues-aware Feature Concentration Transformer Plus (TransIFC+) (H. Liu et al., 2023) reach 94.7% accuracy, which is 0.4% lower than CFCANet. TransFG (He et al., 2021) reports 94.8% accuracy and some regularization techniques can be employed to overcome the overfitting issue brought by PSM. Data Augmented Dual-Attention Interactive Network (DADAINet) (Zhu & Li, 2022) utilizes dual attention which comprises LSTM and MHSA to recalibrate the feature responses based on their importance. The resultant feature maps then undergo the Channel Interaction and Local Feature Fusion module for the generation of more discriminative features. Although DADAINet achieves 94.9% accuracy, the network size is huge as MHSA requires a lot of trainable parameters to perform linear projection on feature maps with high channel counts. Multi-Branch and Multi-Scale Learning Network (MMALNet) (F. Zhang et al., 2021) attains 95.0% accuracy using the sliding window-based part localization technique that is time-consuming. PMG (Du et al., 2020) learns scale-aware features through the jigsaw puzzle learning strategy. Although its performance is on par with CFCANet, it is computationally costly due to the need for multiple forward passes. EfficientNetv2 (Tan & Le, 2021) is a novel CNN architecture proposed recently to reach a better speed-accuracy trade-off. EfficientNetv2-L (112M) produces 95.1% accuracy but it is three times larger than CFCANet (34.1M). PSDPNet (Guo et al., 2022) is a weakly supervised part-based network that contains a raw image, object and part branches. Setting the number of vehicle parts to be identified explicitly compromises the learning since vehicles under different viewpoints may present a different number of distinctive parts.

Table 6.4 illustrates the network performances reported on Car-FG3K (Wu et al., 2022) based on 224×224 resolution without using bounding box information. Since these networks are not evaluated on Car-FG3K in the original works, the training following the training methodology of CFCANet as described in Chapter 6.3.2 is performed. In comparison with ResNet50 (He et al., 2016), the multi-scale features generated by CFCANet ($N_{CFCA} = 3, g = 32$ and $N_d = 3$) raise the performance by 1.9%, reporting 86.2% accuracy. Under the limited number of training images, the performance of TransFG (He et al., 2021) is less favorable i.e. 78.2% due to the lack of inductive bias. MMALNet (F. Zhang et al., 2021) delivers 84.9% accuracy, which is 1.3% lower than CFCANet. For ResNet50-based LRAU (Boukerche & Ma, 2021), it grows from 25.2M to 28.6M and 84.3M when the number of classes increases from 196 (Stanford Cars) to 431 (CompCarsWeb) and 1,892 (Car-FG3K). This implies that the LRAU unit has low scalability to a large number of classes since the number of parameters exhibits quadratic growth with an increasing number of classes. LRAU reports 84.9% accuracy, which is 1.3% lower than CFCANet with 36.4M parameter count. Furthermore, Cross-X Learning (Luo et al., 2019) which consolidates multi-scale features through FPN (Lin et al., 2017a) achieves 85.0% accuracy. Although PMG (Du et al., 2020) and CFCANet are equally competitive on the Stanford Cars dataset, CFCANet demonstrates better representation learning when the training images are scarce where it outperforms PMG by 0.9% in terms of accuracy on the Car-FG3K dataset.

Table 6.4: Performance Benchmarking on Car-FG3K

Reference	Backbone	Accuracy
ResNet50 (He et al., 2016)	-	84.3%
TransFG (He et al., 2021)	ViT_B_16	78.2%
MMALNet (F. Zhang et al., 2021)	ResNet50	84.9%
LRAU (Boukerche & Ma, 2021)	ResNet50	84.9%
Cross-X Learning (Luo et al., 2019)	ResNet50	85.0%
PMG (Du et al., 2020)	ResNet50	85.3%
CFCANet	ResNet50	86.2%

Table 6.5 portrays the works that have the performance validated on CompCarsSV (Yang et al., 2015) based on 224×224 resolution. CFCANet ($N_{CFCA} = 3, g = 32$ and $N_d = 3$) achieves compelling performance, amounting to 99.0% accuracy. The works by Biglari et al. (2017a) and Nazemi et al. (2020) revolve around feature-based domains where handcrafted features are used to fit the machine learning classifiers. To improve feature robustness, Nazemi et al. (2020) utilized LLC (Wang et al., 2010) as a feature encoding layer to transform the raw features into mid-level features. Their networks report 97.5% and 98.4%, respectively. CFCANet also outperforms AlexNet (Krizhevsky et al., 2012), Overfeat (Sermanet et al., 2013) and GoogLeNet (Szegedy et al., 2015) by a significant margin based on the results reported by Yang et al. (2015). Coarse-to-Fine CNN (Fang et al., 2016) reports 98.6% accuracy but it has minimal tolerance against viewpoint variations. Q. Zhang et al. (2018) presented a Lightweight CNN (LWCNN) based on VGG16 (Simonyan & Zisserman, 2014). The proposed changes result in performance degradation from VGG16 and it reports 98.7% accuracy. RAU (Ma & Boukerche, 2020) produces 98.8% accuracy which is slightly lower than CFCANet. In the Feature Fusion-based Car Model Classification Network (FF-CMNet) (Yu et al., 2018), face and non-face vehicle regions are processed separately by different CNNs due to the difference in part significance. Feature fusion is then performed to combine both deep features and 98.9% accuracy is reported. Multi-Agent Systems (MAS) (Amirkhani & Barshooi, 2022) is a part-based ensemble framework consisting of several CNNs. Upon detecting the headlight, upper grill, fog light and bumper using You Only Learn One Representation (YOLOR) (C.-Y. Wang et al., 2021), image processing techniques follow to further refine the Region of Interest (ROI) before using them to fit CNNs. MAS delivers 98.9% accuracy but the proposed technique requires vehicle frontal images so that all the ROIs are available.

Table 6.5: Performance Benchmarking on CompCarsSV

Reference	Backbone	Accuracy
Latent SVM and Cascade Classifier (Biglari et al., 2017a)	-	97.5%
AlexNet (Yang et al., 2015)	-	98.0%
Overfeat (Yang et al., 2015)	-	98.3%
GoogLeNet (Yang et al., 2015)	-	98.4%
Dense-SIFT and LLC (Nazemi et al., 2020)	-	98.4%
Coarse-to-Fine CNN (Fang et al., 2016)	CaffeNet	98.6%
LWCNN with Combined Learning Strategy(Q. Zhang et al., 2018)	VGG16	98.7%
RAU (Ma & Boukerche, 2020)	ResNet50	98.8%
FF-CMNet (Yu et al., 2018)	-	98.9%
MAS (Amirkhani & Barshooi, 2022)	ResNet101, VGG19, Xception, DenseNet201	98.9%
CFCANet	ResNet50	99.0%

For Mohsin-VMMR (Ali et al., 2022), there is no existing network that has been evaluated on this recently published dataset. Nevertheless, the dataset serves as an important performance indicator since it contains multi-view surveillance images that are relevant to the actual case scenario. As shown in Table 6.6, upon incorporating the CFCA module into ResNet50 (He et al., 2016) ($N_{CFCA} = 3, g = 32$ and $N_d = 3$), the classification accuracy reaches 96.9% and the percentage of improvement is 0.9%.

Table 6.6: Performance Benchmarking on Mohsin-VMMR

Reference	Backbone	Accuracy
ResNet50 (He et al., 2016)	-	96.0%
CFCANet	ResNet50	96.9%

The performance of CFCANet is further examined through breakdown analysis as portrayed in Table 6.7 and the corresponding confusion matrix is shown in Figure 6.3. Table 6.7 and Figure 6.3 unravel that CFCANet achieves commendable prediction for all classes but the 20th class. The true prediction for the 20th class which is Suzuki Carry (mini truck) accounts for 33.3% and this is largely due to insufficient training images. The number of training images for Suzuki Carry is the least among all classes, it stands

at 13 images only. Apart from that, it is realized that all the false predictions fall into the Suzuki Highroof (van). This is due to the intra-make ambiguity as shown in Figure 6.4. The accuracy of Suzuki Carry can be improved by collecting more training images.

Table 6.7: Performance Breakdown Analysis of CFCANet on Mohsin-VMMR

Make	FAW	Toyota		Honda	
Model	XPV	Premio	Prado	Civic'15	Civic'18
Acc	100	75.0	100	100	100
Make	Toyota	Suzuki	FAW	Toyota	Kia
Model	Landcruiser	Alto'07	V2	Prius	Sportage
Acc	100	100	100	100	100
Make	Suzuki	Toyota	Honda		Toyota
Model	Alto'10	Corolla'00	BRV	Vezel	Hiace'00
Acc	100	100	100	100	100
Make	Toyota		Honda	Suzuki	Toyota
Model	Aqua	Passo	City Aspire	Khyber	Corolla'16
Acc	78.9	100	96.2	100	98.5
Make	Suzuki	Honda	Toyota		Daihatsu
Model	Carry	City'94	Hiace'12	Fortuner	Core
Acc	33.3	100	94.1	100	100
Make	Daihatsu	Suzuki	Honda	Toyota	Suzuki
Model	Mira	WagonR'15	Grace	Axio	Highroof
Acc	100	96.3	80.0	100	100
Make	Suzuki		Honda	Suzuki	
Model	Mehran	Every	City'00	Alto'19	Liana
Acc	100	80.0	87.5	85.7	87.5
Make	Toyota	Toyota	Suzuki	Toyota	Daihatsu
Model	Vigo	Corolla'11	Swift	Vitz'10	Hijet
Acc	92.3	100	96.6	100	100
Make	Suzuki	Toyota	Honda		Suzuki
Model	Margala	Vitz	Civic'94	Civic'07	Cultus'19
Acc	100	100	100	100	100
Make	Honda	Toyota	Suzuki	Average Accuracy=95.0%	
Model	Civic'05	Corolla'07	Cultus'18		
Acc	87.5	90.0	100		

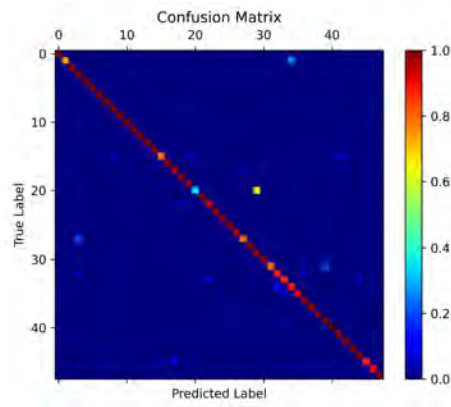


Figure 6.3: Confusion Matrix of CFCANet for VMMR-Mohsin



Figure 6.4: Suzuki Carry (Top Row) and Suzuki Highroof (Bottom Row)

6.5.2 Qualitative Analysis

To intuitively compare the baseline (ResNet50) and CFCANet in terms of their abilities to capture the distinctive vehicle parts, the top-level feature maps of both networks are visualized by applying Gradient-Weighted Class Activation Mapping (Grad-CAM) (Selvaraju et al., 2017). Grad-CAM is a visualization technique that utilizes gradient information to highlight the prominent region. The region which shows a high gradient corresponds to the significant region that plays a vital role in the VMMR task.

Table 6.8 illustrates the top-level feature maps visualized using Grad-CAM for 5 vehicle models. The top-level feature maps from ResNet50 and CFCANet are $I_{CB,4}$ and $I_{FI,4}$, respectively. Additional images of the same vehicle models but with different viewpoints are also selected to demonstrate the learning of the proposed network under multi-view scenarios. It is important to note that the class-specific discriminative region is highlighted in red whereas the less discriminative region is highlighted in blue.

By comparing both heatmaps, it is observed that the heatmap of CFCANet covers a wider region. For instance, the baseline network pays attention to the headlights given the frontal view of Land Rover Range Rover. Although the design of the headlights is unique (Amirkhani & Barshooi, 2022), focusing on them alone increases the complexity of the VMMR task. On the contrary, CFCANet enlarges the focused region to include the whole frontal region but the windscreen. On top of headlights, it takes the grill, bumper, fog light and vehicle logo into account and these attributes present more key information for the network to perform classification. Referring to the rear view of the Audi S6, the baseline network attends to the backlight and leaves out the distinctive vehicle logo whereas CFCANet identifies the backlight, vehicle logo and car boot as the prominent clues. Looking at the side view of the Rolls-Royce Phantom, CFCANet also identifies more crucial vehicle parts as compared to the baseline network. As most of the details are occluded, the whole vehicle is picked up by CFCANet to elevate recognition ability.

To encapsulate, the focused region of CFCANet is more holistic and selective where the vital vehicle parts are considered and the ordinary vehicle parts are discarded. This is driven by the CFCA module which unifies the fine-grained structural information from low-level feature maps and semantically meaningful information from high-level feature maps to present comprehensive multi-scale feature maps that make the class boundary more separable.

The learned feature embeddings between ResNet50 and CFCANet are compared by performing dimension reduction using T-distributed Stochastic Neighbour Embedding (TSNE) (van der Maaten & Hinton, 2008). Specifically, the deep features from the penultimate layer of both networks are extracted and they are reduced into two-dimensional vectors. Figure 6.5 depicts the learned features upon being projected into two-dimensional space. It is worth noting that each data point represents the features of an image and it is coloured based on the label.

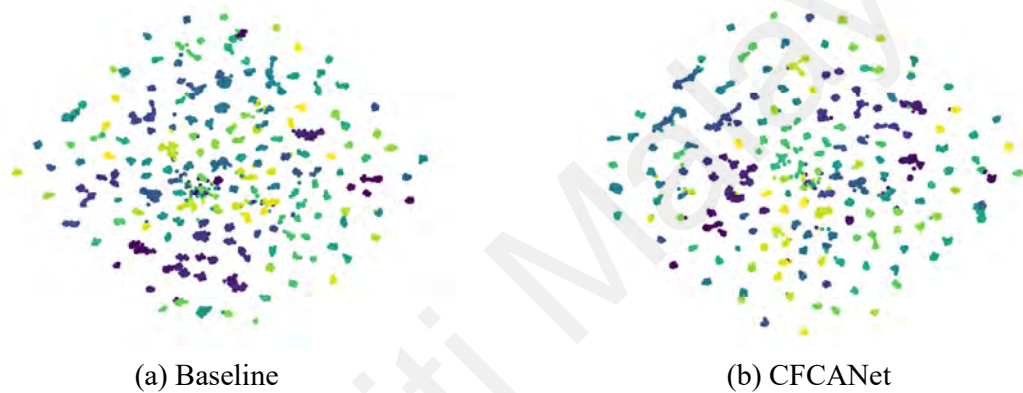


Figure 6.5: Visualization of Feature Embeddings

Based on Figure 6.5 (a), it is observed that there is a significant overlap between classes. Since the decision boundary is vague, the classification accuracy of the baseline is lower. On the contrary, the data points in Figure 6.5 (b) are clustered relatively well into their respective classes. Although not all the classes are fully separable from one another, the illustration shows that CFCANet is able to learn a more compact representation where the intraclass variance is lower.

Table 6.8: Visualization of Feature Maps Using Grad-CAM

Audi S6		Hyundai Sonata		Volkswagen Golf		Rolls-Royce Phantom		Land Rover Range Rover	
Baseline	CFCANet	Baseline	CFCANet	Baseline	CFCANet	Baseline	CFCANet	Baseline	CFCANet

6.5.3 Ablation Study

To justify the hyperparameter of choice for the CFCA module, an ablation study is conducted on CompCarsWeb and Stanford Cars. The results are deliberated in this section. Since the datasets are imbalanced, precision, recall and f1-score are reported in Table 6.9 and Table 6.10. The number of parameters and floating point operations (FLOPs) are also attached to gauge the required computational cost.

Table 6.9: Ablation Study on CompCarsWeb

Exp.	N_{CFCA}	g	N_d	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	#Params (M)	GFLOPs
Baseline	-	-	-	96.9	97.1	96.6	96.7	24.4	4.1
A	1	32	3	97.6	97.7	97.4	97.5	31.3	4.5
	2	32	3	97.6	97.7	97.4	97.5	33.0	4.8
	3	32	3	98.0	98.1	97.8	97.9	33.4	5.1
	4	32	3	97.7	97.8	97.5	97.6	33.5	5.5
B	3	8	3	97.7	97.8	97.5	97.6	43.9	6.3
	3	16	3	97.9	98.0	97.7	97.8	36.9	5.5
	3	32	3	98.0	98.1	97.8	97.9	33.4	5.1
C	3	64	3	97.9	98.0	97.7	97.8	31.7	5.0
	3	32	1	97.6	97.7	97.3	97.5	31.1	4.9
	3	32	2	97.7	97.8	97.4	97.5	32.2	5.0
	3	32	3	98.0	98.1	97.8	97.9	33.4	5.1
	3	32	4	97.9	98.0	97.7	97.8	34.6	5.3

Table 6.10: Ablation Study on Stanford Cars

Exp.	N_{CFCA}	g	N_d	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	#Params (M)	GFLOPs
Baseline	-	-	-	92.9	93.2	92.8	92.8	23.9	16.5
A	1	32	3	94.6	94.7	94.5	94.5	30.8	17.9
	2	32	3	94.5	94.6	94.5	94.4	32.5	19.2
	3	32	3	94.6	94.8	94.5	94.6	32.9	20.6
	4	32	3	94.5	94.6	94.5	94.4	33.0	21.8
B	3	8	3	94.3	94.4	94.2	94.2	43.4	25.3
	3	16	3	94.5	94.6	94.4	94.5	36.4	22.1
	3	32	3	94.6	94.8	94.5	94.6	32.9	20.6
C	3	64	3	94.4	94.4	94.3	94.3	31.2	19.8
	3	32	1	94.7	94.8	94.6	94.6	30.6	19.5
	3	32	2	94.8	94.9	94.8	94.7	31.8	20.1
	3	32	3	94.6	94.8	94.5	94.6	32.9	20.6
	3	32	4	95.1	95.1	95.0	95.0	34.1	21.1

The objective of the CFCA module is to combine an optimum number of feature maps with different scale information to form the final features that are rich in both local and global contexts. Based on Experiment A from Table 6.9 and Table 6.10, setting $N_{CFCA} = 1$ gives decent performances on CompCarsWeb and Stanford Cars, which are 97.6% and 94.6%, outperforming the baseline ResNet50. When more feature maps from the lower level and top-most level are merged, more granular information is obtained and significant performance gain can be seen. By setting $N_{CFCA} = 3$, CFCANet renders the best performance. For instance, CFCANet reports 98.0% accuracy on CompCarsWeb. However, setting $N_{CFCA} = 4$ leads to performance degradation. The reason is utilizing too many low-level feature maps brings in noisy information where they focus on non-discriminative vehicle parts (Gao et al., 2021). Hence, N_{CFCA} should be set with caution for optimal performance.

Grouped convolution plays a significant role in the design of the CFCA module where it minimizes the number of parameters and ensures moderate network size. Moreover, adopting grouped convolution ensures diversified learning since the convolution kernels are less correlated across the groups. Referring to Experiment B from Table 6.9 and Table 6.10, the starting value of g is set as 8 to avoid CFCANet from being excessively large and the network growing into having more than 40M parameters. Increasing g from 8 to 32 brings a positive impact on both performance and network size. The parameters of CFCANet are reduced by 24% and 0.3% accuracy improvement is seen on both datasets. When g is set to 64, there is a minor drop in accuracy. For the Stanford Cars dataset, CFCANet drops 0.2% to 94.4% when g increases from 32 to 64. It is assumed that setting an extremely large value for g limits the inter-channel information exchange and this refrains the network from learning more discriminative features (Xie et al., 2020).

The CFCA module utilizes a dilation rate to expand the receptive field of convolution kernels. Based on Experiment C from Table 6.9 and Table 6.10, it is concluded that an appropriate N_d is pivotal. On CompCarsWeb, the classification accuracy increases from 97.6% to 98.0% when sweeping from $N_d = 1$ to $N_d = 3$. A similar improvement is observed in the Stanford Cars dataset. The result suggests that a large receptive field is beneficial to the aggregation of key features especially when the key vehicle parts are not adjacent to each other. However, setting $N_d = 4$ degrades the performance of CFCANet on CompCarsWeb but improves further the performance on Stanford Cars. It is construed that the optimal value for N_d is dataset dependent. A large dilation rate deteriorates the performance since it causes spatial inconsistency between the neighboring feature responses (Hamaguchi et al., 2018).

6.5.4 Generalization Study

The compatibility between the CFCA module and existing CNNs is further inspected. The compatibility specifically refers to the generalization ability of the CFCA module on other variants of CNNs. It is quantified in terms of accuracy improvement from the backbone networks and the results are depicted in Figure 6.6 and Figure 6.7. In particular, the experiment is performed using VGG16 (Simonyan & Zisserman, 2014), Inceptionv3 (Szegedy et al., 2016), ResNet50 (He et al., 2016) and DenseNet169 (Huang et al., 2017). For CompCarsWeb, N_{CFCA} , g and N_d are set as 3, 32 and 3, respectively for all networks. For Stanford Cars, N_{CFCA} , g and N_d are set as 3, 32 and 4, respectively for all networks.

The performance gain is significant where 1.4%, 1.0%, 1.1% and 0.7% accuracy improvements are recorded for VGG16, Inceptionv3, ResNet50 and DenseNet169, respectively on CompCarsWeb. On Stanford Cars, the accuracies improvement are 2.2%, 2.6%, 2.2% and 0.3% for VGG16, Inceptionv3, ResNet50 and DenseNet169, respectively. The increment in accuracy for DenseNet169 is the least and this is due to

the dense connection between the convolution layers which is similar to the CFCA module to a certain extent.

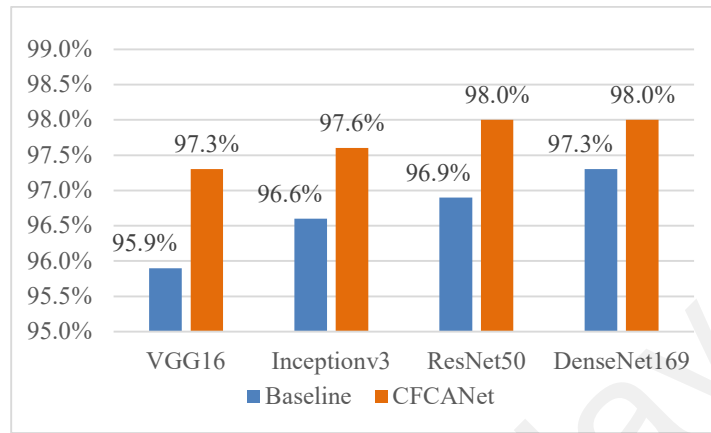


Figure 6.6: Compatibility between CFCA module and Existing CNNs on CompCarsWeb

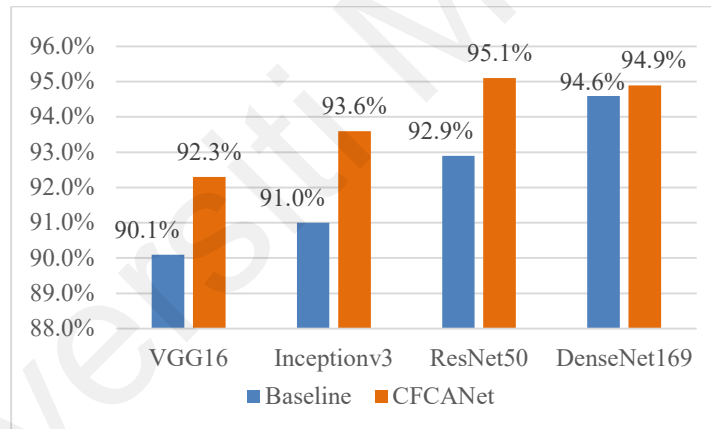


Figure 6.7: Compatibility between the CFCA module and Existing CNNs on Stanford Cars

In terms of the number of parameters, the increment is the least for VGG16, followed by DenseNet169, ResNet50 and Inceptionv3. CFCA module introduces an additional 1M parameters only to VGG16 due to the low channel counts of the feature maps, which is 512 as compared to Inceptionv3's 2048, ResNet's 2048 and DenseNet169's 1664. Inceptionv3 has the largest leap in terms of parameters because of lower g i.e. 16 as compared to other networks ($g = 32$). g is set as 16 for Inceptionv3 since not all the

channel counts are a factor of 32. The number of parameters for Inceptionv3 can be brought down by setting a higher value of g .

6.6 Conclusion

In this chapter, the CFCA module is proposed to fully exploit the pyramidal features from the backbone CNN architectures. Conventionally, the feature embeddings from top-level feature maps are used to model vehicle properties. Although they carry semantically strong information, they lack the spatial and geometric details which are the important cues for various vehicle models. By augmenting a bottom-up path, the CFCA module learns the nonlinear aggregation of multi-scale feature maps to render compact representations that retain both semantic information and fine-grained critical traits. Essentially, dilated convolution which expands the field of view of convolution kernels is adopted to enrich the feature representations before consolidating the scale-specific features together. The proposed framework achieves state-of-the-art results by reporting 98.0%, 95.1%, 86.2%, 99.0% and 96.9% on CompCarsWeb, Stanford Cars, Car-FG3K, CompCarsSV and Mohsin-VMMR, respectively. Additionally, the qualitative analysis depicts the ability of the CFCA module to identify the crucial vehicle parts that are overlooked by the baseline network. Lastly, the high compatibility between the CFCA module and various backbone CNNs such as VGG16, Inceptionv3, ResNet50 and DenseNet169 is also demonstrated where it results in an average 1.1% and 1.8% accuracy improvement based on CompCarsWeb and Stanford Cars, respectively.

The study also reveals that optimizing the receptive field, especially during the FE stage plays a significant role in feature representation learning. For future work, a module that exerts a global receptive field and with modest computational complexity will be studied to exploit the relationship among all the spatial positions.

CHAPTER 7: AUGMENTED-GRANULARITY NETWORK FOR VEHICLE MAKE AND MODEL RECOGNITION

7.1 Introduction

This chapter expounds on a multi-scale feature generation module that is performance-oriented. In particular, an Augmented-Granularity (AG) module that ingests multi-granularity information collected from various pyramid levels is proposed to build comprehensive feature maps. At the individual pyramid level, the AG module refines the features generated by the backbone Convolutional Neural Networks (CNNs) through grouped focus convolution (GFConv). GFConv exercises a broader receptive field via space-to-depth transformation and it allows convolution to be performed in groups for reduced computational complexity. Eventually, all the refined scale-specific features are consolidated via 1×1 convolution where the network learns the relative importance of the hierarchical features. The contributions of this chapter are summarized as follows:

- Propose the AG module that distills low-level structural details and high-level semantic information to form condensed feature maps
- Conduct performance benchmarking and perform the qualitative inspection of the learned feature representations
- Demonstrate the high generalization ability of the AG module on existing CNNs

7.2 Literature Review

Feature-based methods refer to works that use handcrafted features and they spearheaded the development of Vehicle Make and Model Recognition (VMMR) in the early times. The handcrafted features are generated by examining the gradient orientation information or interest points. For instance, the Histogram of Oriented Gradient was leveraged by Manzoor et al. (2019) to derive the raw vehicle attributes. Hsieh et al. (2014) experimented with a novel grid-based solution constructed on Scale-Invariant Feature

Transform (SIFT) (Lowe, 2004) and Speeded Up Robust Features (SURF) (Bay et al., 2008). Since handcrafted features are less robust to external disturbances, several feature encoding schemes were proposed to transform them into mid-level features. Siddiqui et al. (2016) quantized SURF features through Bag of Features (BoF) and utilized them to differentiate 29 vehicle models. Extending from BoF, Jamil et al. (2020) introduced the Bag of Expressions (BoE) to mitigate information loss during quantization and improve viewpoint robustness. Furthermore, Nazemi et al. (2020) considered locality constraints by encoding the raw dense SIFT features with Locality-constraint Linear Coding (LLC) (Wang et al., 2010). Although the improvement induced by feature encoding schemes is motivating, choosing the optimum feature extractor algorithm requires in-depth experience.

Part-based methods enhance the VMMR performance by imposing localization techniques to focus on the vehicles and suppress the irrelevant background noise that disturbs the learning of discriminative features. Ghassemi et al. (2019) introduced the Wide ResNet50-based Multiscale Attention Windows Network (MAWNet) where the Spatial Transformer Network (STN) (Jaderberg et al., 2015) is utilized to learn affine transformation to extract the distinctive vehicle parts in a weakly-supervised manner. Their proposal requires a complex training pipeline since STN and Wide ResNet50 (Zagoruyko & Komodakis, 2016) are required to undergo training separately to ensure convergence of loss function. In Multilayer Bilinear Pooling Network (MLBPNet), Ming Li et al. (2022) produced the vehicle mask through the binarization of feature maps from upper pyramid levels and the vehicle information is subsequently characterized by high-order statistics via the MLBP module. Due to the huge embedding size i.e. 8192-dimensional vector produced by the MLBP module, their network has low scalability to datasets with a high number of classes. Progressively Sampling Discriminative Parts Network (PSDPNet) (Guo et al., 2022) also localizes the vehicle together with the

prominent parts by observing the peak responses on feature maps upon weighing them via class activation mapping and spatial correlation matrix. Nevertheless, their solution may degrade significantly under multi-view scenarios due to inconsistency in discriminative parts sampling.

Attention-based category unravels the works that perform feature map refinement based on information relevancy. It differs from the part-based methods such that less weightage is allocated to the regions that seem trivial instead of eradicating them. In the Convolutional Attention Model (ConvAM) (Yu et al., 2020), the feature maps from early convolution layers are recalibrated by considering the inter-channel dependency via Long Short-Term Memory (LSTM). A Convolutional-LSTM attention module is also applied to the top-level feature maps to model the temporal and spatial-temporal variations. The proposal raises the performance of the backbone network but it is accompanied by a huge increase in the number of parameters. Ma and Boukerche (2020) proposed Recurrent Attention Unit (RAU) where attention masks are refined in stages to highlight the discriminative features. RAU can be ameliorated further to improve computational efficiency. Attentive Pairwise Interaction Network (APINet) (Zhuang et al., 2020) suggests a unique cross-image learning strategy that learns the key object parts through a compare and contrast process. However, the training process requires a careful design of image pair construction strategy to achieve convergence of loss function.

Works categorized under the novel backbone category present new architectures that advance the classification performance. EfficientNetv2 (Tan & Le, 2021) is a novel CNN evolved from EfficientNet (Tan & Le, 2019) where the building blocks are optimized for better computational efficiency. A progressive learning tactic is also applied to tune the regularization strength along with the image resolution to improve the learning process. In addition, Zhuang Liu et al. (2022) modernized the CNN architectures by improvising

ResNet (He et al., 2016). The modifications include alteration of macro and micro designs, utilization of grouped convolution, and inverted bottleneck as well as large convolution kernel size. The network dubbed ConvNeXt is notorious as it improves the classification performance and maintains the simplicity and efficiency of CNNs. Since they are designed for general image classification tasks, their performances will rise further when they incorporate elements tailored for fine-grained vehicle classification. Inspired by the success of transformers in addressing Natural Language Processing (NLP) tasks, a transformer-based architecture, namely Vision Transformer (ViT), is proposed by Dosovitskiy et al. (2020). ViT flattens the 2-dimensional image into a sequence of image tokens and processes them with multiple encoder blocks consisting of Multi-Head Self-Attention (MHSA) and Feed Forward Network (FFN). Although ViT can produce more holistic feature representation by tracking long-range dependencies, the class embedding token fails to encapsulate the learned information from all tokens and hence the performance is less competitive (Kang et al., 2022; Touvron et al., 2021; Yuan et al., 2021). Another notable transformer architecture is the Shifted Windows (Swin) Transformer (Z. Liu et al., 2021) which implements a shifted windowing scheme. Despite successfully reducing quadratic complexity to linear, the local window-based MHSA debilitates the ability to model global dependencies (Qin et al., 2022).

Multi-scale features encompass the spatial and semantic information that is essential for accurate VMMR from the hierarchical feature maps. G. Wang et al. (2021) implemented Pyramid Convolution (PyConv) that employs convolution kernels of multiple sizes. Despite aggregating features from a mixture of receptive field sizes, the resultant representation lacks the local granular details since the classification logits are still inferred from the top-level feature maps. In Progressive Multi-Granularity (PMG) network (Du et al., 2020), the features at every scale level are exploited where the prediction is collectively determined by the classification heads at each pyramid level. It

brings remarkable classification performance but the potential of multi-scale features is not fully harnessed as the features are being treated independently at respective scale levels. The proposed AG module overcomes this by forging interaction among the scale-varying components to promote the information exchange and allow adaptive weighing between coarse- and fine-grained components. Jung et al. (2017) proposed a Feature Covariance Attention (FCA) module that propagates the scale-specific information to the immediate next pyramid level. This practice results in the domination of large-scale components over small-scale components. Instead of passing the information layer-to-layer, the AG module augments a lateral path to allow direct integration of cross-layer information and this prevents the loss of subtle details. Feature Pyramid Network (FPN) (Lin et al., 2017a) is also a popular technique for realizing cross-scale information transfer through upsampling and addition operations within the adjacent scales. In the AG module, the simple feature transfer method of FPN is improvised by merging information from various scale levels in a non-linear manner based on information saliency.

7.3 Methodology

Fine-grained vehicle classification is a CV application that mines tiny but crucial vehicle characteristics to achieve differentiation at a granular level. Leveraging top-level feature maps alone is detrimental since the perception capability towards the vehicle details is limited. The low-level feature maps can provide complementary spatial context information to achieve a more holistic understanding. The AG module is designed to achieve this purpose where it improves the performance of the backbone network through encapsulation of the local and global features within the final feature representations.

7.3.1 Augmented-Granularity Module

Figure 7.1 illustrates a backbone CNN that is integrated with the AG module. The AG module aims to achieve two outcomes. Firstly, it improves feature expressiveness by

performing feature refinement using convolution kernels with an expanded receptive field during the Feature Extraction (FE) stage. Secondly, it renders a balanced mix between the feature maps from all scale levels by dynamically fusing them based on importance in the Feature Integration (FI) stage. Overall, the AG module elevates the VMMR performance by considering both microscopic and macroscopic features.

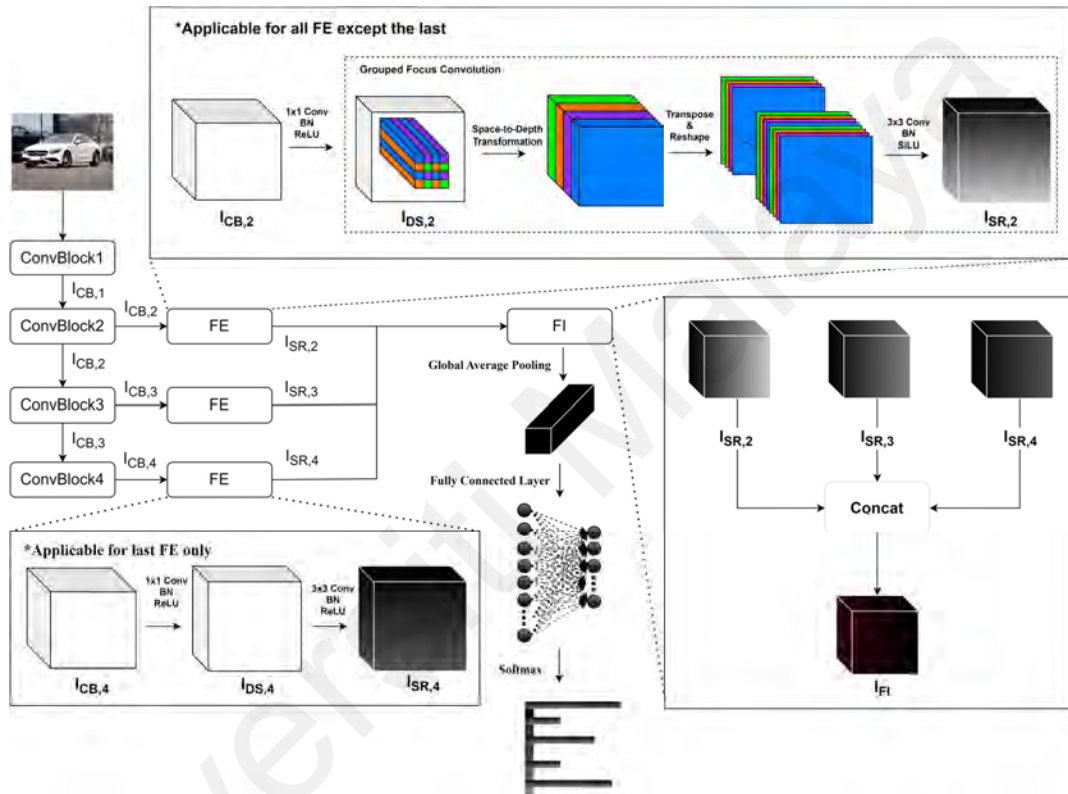


Figure 7.1: Augmented-Granularity Network

Given an input image $I_i \in \mathbb{R}^{H \times W \times C}$ where H, W and C represent height, width, and number of channels, respectively, it first passes through N_{CB} convolutional blocks (CBs) to produce hierarchical feature maps where $N_{CB} = 4$ for most of the CNNs. The resultant feature maps, denoted as $\{I_{CB,i}\}_{i=1}^4 \in \mathbb{R}^{H_{CB,i} \times W_{CB,i} \times C_{CB,i}}$, carry low-level, mid-level and high-level information that is significant to the prediction task.

Targeting to retain the scale-specific components, the AG module receives $I_{CB,i}$ as input and processes them further in the FE stage. The FE stage serves the purpose of

standardizing the channel dimension and spatial resolution of all the pyramidal features for the subsequent merging operation in the FI stage. To keep the AG module lightweight in terms of the number of parameters and floating point operations (FLOPs), $I_{CB,i}$ has its channel dimension reduced through 1×1 convolution to form $I_{DS,i} \in \mathbb{R}^{H_{CB,i} \times W_{CB,i} \times C_{FE}}$. The depth standardization (DS) operation is represented as

$$I_{DS,i} = Conv_{1 \times 1, C_{FE}}(I_{CB,i}) \quad (7.1)$$

where $C_{FE} = \lfloor C_{CB,4}/4 \rfloor$ is the resultant channel dimension. The purpose of standardizing the channel dimension is to restrain the dominance of information at any scale level over the other in the FI stage. After the convolution layer, batch normalization (BN) and rectified linear unit (ReLU) follow to prevent covariate shift and inject non-linearity.

Subsequently, the FE stage implements 3×3 convolution with an enlarged receptive field to improve the feature expressiveness. A common approach to expanding the receptive field is through dilated convolution. However, adjusting the dilation rate beyond 1 causes information loss when the pixels at the boundary of the feature maps are discarded. To bridge the gap, GFConv is proposed and the PyTorch-style pseudocode is summarized in Algorithm 7.1.

Specifically, bilinear interpolation is first carried out to enlarge feature maps if they are odd in size. The space-to-depth transformation operation then transforms the spatial information into the channel dimension. This doubles the effective receptive field size as compared to that of the conventional convolution layer. For instance, a 3×3 kernel now exerts an effective window size of 6×6 . Nevertheless, the transformation operation comes with additional computational burdens since the channel dimension is now increased by 4. To alleviate this, the flexibility of performing grouped convolution is enabled. As a transition step to grouped convolution, the transpose and resize operations are

implemented to ensure every feature map is interleaved with the feature map from the other spatial positions. This step encourages diversified learning where the information exchange happens in cross-spatial positions.

Algorithm 7.1 Pseudocode for Grouped Focus Convolution

Batch Size: B ; Channel Dimension: C ; Height: H ; Width: W ; Kernel Size: k ; Stride: s ; Group: g , Padding: p , Batch Normalization: BN, Sigmoid Linear Unit: SiLU
Input shape: (B, C, H, W)
Output shape: $(B, C, H/2s, W/2s)$

```

1 def init ():
2     conv = Conv(4*C, C, k=3, s, g, p=0)
3     bn = BN(C)
4 def forward (x):
5     B, C, H, W = x.size()
6     if H%2 !=0: H += 1
7     if W%2 !=0: W += 1
8     x = interpolate(x, size=(H,W), mode='bilinear')
9     x = stack((x[..., ::2, ::2], x[..., 1::2, ::2], x[..., ::2, 1::2],
10              x[..., 1::2, 1::2]), dim=1)
11    x = x.transpose(1, 2)
12    x = x.reshape(B, 4*C, H//2, W//2)
13    x = conv(x)
14    x = bn(x)
15    return SiLU(x)

```

Another accompanying effect of the GFConv is the downsampling behavior where the input feature maps are downsized by half even though stride s is set as 1. This behavior aligns with the interest in carrying out spatial reduction (SR) to minimize the memory footprint. In particular, the resultant spatial size is set as $H_{CB,4}$ and $W_{CB,4}$. To achieve this outcome, s is chosen as 4, 2 and 1 when GFConv is applied on $\{I_{DS,i}\}_{i=1}^{i=3}$. The focus convolution layer with group convolution capability is described as follows:

$$I_{SR,i} = GFConv_{3 \times 3, C_{FE}, s, g, p}(I_{DS,i}) \quad (7.2)$$

where g and $p = 0$ are the number of groups and padding, respectively.

For $I_{DS,4}$, the conventional 3×3 convolution is adopted since no downsampling operation is desired at this level. The operation is denoted as

$$I_{SR,4} = Conv_{3 \times 3, C_{FE}, s, g, p}(I_{DS,4}) \quad (7.3)$$

where $s = g = p = 1$ and it is followed by BN and ReLU operations.

After the FE stage refines the scale-specific feature maps, the FI stage exploits the inter-scale relationship by promoting information propagation across multiple pyramid levels. This is made viable by consolidating the feature maps that carry various degrees of structural and semantic information. The consolidation process should be versatile such that it is based on the relative importance between the low-level and high-level feature maps. Aligned with this objective, $\{I_{SR,i}\}_{i=4-N_{FE}+1}^4$ are concatenated along the channel axis and 1×1 convolution is applied to perform cross-channel pooling. The operation performed during the FI stage is described as

$$I_{FI} = Conv_{1 \times 1, C_{FI}} \left(\left[\{I_{SR,i}\}_{4-N_{FE}+1}^4 \right] \right) \quad (7.4)$$

where N_{FE} is the number of feature pyramids to use, $[\bullet]$ denotes channel concatenation, $C_{FI} = N_{FE} \times C_{FE}$ and the convolution operation is followed by BN and ReLU.

To summarize, by sending $I_{CB,i}$ to the AG module, a multi-granularity feature map, I_{FI} , is deduced. A classification head is attached to it to produce classification logits.

$$I_{FI}^{GAP} = GAP(I_{FI}) \quad (7.5)$$

$$Logits = FC_L(BN(I_{FI}^{GAP})) \quad (7.6)$$

where $GAP(\bullet)$, and FC_L are global average pooling and fully connected layer with L output neurons, respectively.

7.3.2 Augmented-Granularity Network

The AG module is modular and can be easily integrated with the existing CNNs to improve the feature learning process. 4 CNNs, namely VGG16 (Simonyan & Zisserman, 2014), Inceptionv3 (Szegedy et al., 2016), ResNet50 (He et al., 2016) and TResNet-L (Ridnik et al., 2021a) are used to justify this. Generally, the network, upon integration with the AG module, is called AGNet. It is also important to note that the insertion of the AG module at high-level takes precedence over low-level. For instance, given $N_{CB} = 4$, setting $N_{FE} = 3$ will establish lateral connections from $\{I_{CB,i}\}_{i=2}^4$.

VGG16 (Simonyan & Zisserman, 2014) lays the foundation for modern CNN architectures. It replaces the large-size convolution kernels of AlexNet (Krizhevsky et al., 2012) with consecutive convolution operations of smaller kernel sizes to offer the same receptive field size. This brings a significant reduction in the number of parameters and allows the network to grow deeper to learn more discriminative features. In the experiment, VGG16 is modified so that it is compatible with input with dynamic resolution. In particular, after the top-level feature maps, GAP is applied and the parameter-heavy three-layer FC is replaced with a single-layer FC. For the construction of VGG16-AG, the feature maps produced by the last four maximum pooling layers are identified as $\{I_{CB,i}\}_{i=1}^4$. Based on $224 \times 224 \times 3$ I_i , the sizes of $\{I_{CB,i}\}_{i=1}^4$ are $128 \times 128 \times 64$, $56 \times 56 \times 128$, $28 \times 28 \times 256$, $14 \times 14 \times 512$ and $7 \times 7 \times 512$.

Inceptionv3 (Szegedy et al., 2016) revamps the Inception module proposed by its predecessor, which is GoogLeNet (Szegedy et al., 2015). The refined Inception module performs convolution in 4 parallel branches with reduced computational costs. The

expensive 5×5 convolution kernels are substituted with 2 consecutive 3×3 convolution kernels to maintain the receptive field size. The Inception module is made more lightweight by factorizing the convolution where $k \times k$ convolution is broken down into $1 \times k$ and $k \times 1$ convolution. Inceptionv3 also features an efficient grid size reduction strategy without having a representation bottleneck. Based on $299 \times 299 \times 3$ I_i , feature maps $71 \times 71 \times 192$, $35 \times 35 \times 288$, $17 \times 17 \times 768$ and $8 \times 8 \times 2048$ are identified as $\{I_{CB,i}\}_{i=1}^4$.

ResNet (He et al., 2016) follows the design philosophy of VGG16 (Simonyan & Zisserman, 2014) where more expressive features can be learned by growing the CNN in depth. However, the complication is the vanishing gradient problem, especially at the early layers. ResNet resolves this issue through the introduction of the shortcut connection to allow the gradients to be backpropagated more effectively without adding any computational complexity. Based on $224 \times 224 \times 3$ I_i , $\{I_{CB,i}\}_{i=1}^4$ are the feature maps produced by every bottleneck building block.

Ridnik et al. (2021a) modified the ResNet architecture to provide a boost to the classification performance as well as retain the GPU training and inference efficiency. The proposed refinements such as SpaceToDepth Stem, Anti-Alias Downsampling, In-Place Activated BatchNorm, Novel Block-Type Selection and Optimized Squeeze-and-Excitation layers make TResNet significantly better than its baseline. $\{I_{CB,i}\}_{i=1}^4$ are the output of the building blocks of TResNet.

7.3.3 Loss Function

The notable cross entropy loss with label smoothing coefficient \mathcal{E} (Szegedy et al., 2016) is employed as the loss function. Setting $\mathcal{E} > 0$ injects a uniformly distributed noise to the labels. Hence, instead of training the network on hard labels, the network is

trained on soft labels and this precludes the network from overfitting. The cross entropy with the label smooth coefficient is given as

$$Loss_{LS} = -\frac{1}{N_{Train}} \sum_{i=1}^{N_{Train}} \sum_{l=1}^L (1 - \varepsilon) y_{i,l} \log p_{i,l} + \frac{\varepsilon}{(L - 1)} y_{i,l} \log p_{i,l} \quad (7.7)$$

where $y_{i,l}$ is one-hot encoded ground truth, $p_{i,l}$ is the probability distribution predicted by the network and N_{Train} is the number of training samples.

The center loss (Wen et al., 2016) is also incorporated into the loss function to reduce the intraclass variance and interclass similarity of the embeddings in the feature space. The center loss penalizes the network based on the Euclidean distance between the image samples and class feature center matrix $C^t \in \mathbb{R}^{L \times C_{FI}}$. It is computed as

$$Loss_{Center} = \frac{1}{N_{Train}} \sum_{i=1}^{N_{Train}} \|I_{FI,i}^{GAP} - C_{y_i}^t\|_2^2 \quad (7.8)$$

where $I_{FI,i}^{GAP}$ is the feature vector for i^{th} training image, $C_{y_i}^t$ is the feature center vector for the class y_i at iteration t and $\|\bullet\|$ is the Euclidean norm. For every iteration, C^t is updated and its representation power as the class feature center becomes increasingly stronger. Overall, the loss function used in the experiment is denoted as follows:

$$Loss = Loss_{LS} + \alpha_{Center} Loss_{Center} \quad (7.9)$$

where α_{Center} is the contribution of center loss.

7.4 Experiments

7.4.1 Datasets

The AG module is validated through 4 public datasets, which are Car-FG3K (Wu et al., 2022), Stanford Cars (Krause et al., 2013), Web-Nature Comprehensive Cars

(CompCarsWeb) (Yang et al., 2015) and VMRRdb (Tafazzoli et al., 2017). They present different complexity levels due to the number of classes and the size of the datasets.

Car-FG3K (Wu et al., 2022) is a dataset published recently. It contains 10,676 training and 10,661 testing images. Despite being moderate in size, it covers 1,892 classes. The mean training image count for a class is 5.6, which is significantly lower than Stanford Cars (Krause et al., 2013) (41.6) and CompCarsWeb (Yang et al., 2015) (84.6). The vehicle images appear in various viewpoints and no bounding box annotation is provided.

Stanford Cars (Krause et al., 2013) is a notable fine-grained visual classification dataset. Having approximately 50:50 train-test split ratio, it contains 196 classes. There is also diversity in terms of viewpoints variations which include front, rear, side, front-side and rear-side. The existence of multi-view images contributes to the training of a robust network that works under multi-view scenarios.

The size of CompCarsWeb (Yang et al., 2015) is larger than both Car-FG3K (Wu et al., 2022) and Stanford Cars (Krause et al., 2013). It carries a total of 36,456 training and 15,627 testing images for 431 vehicle models. As the name suggests, the images are obtained from various online resources.

VMRRdb (Tafazzoli et al., 2017) is the largest dataset used in the experiment. There is no train-test annotation and bounding box information provided. Among the vehicle models that have an image count larger than 57, stratified sampling based on the 70:30 train-test split ratio is performed. This results in 192,494 training and 82,507 testing images for 497 classes.

Fig. 7.2 shows a few examples of the datasets used in the experiment and a summary of the statistics is tabulated in Table 7.1.

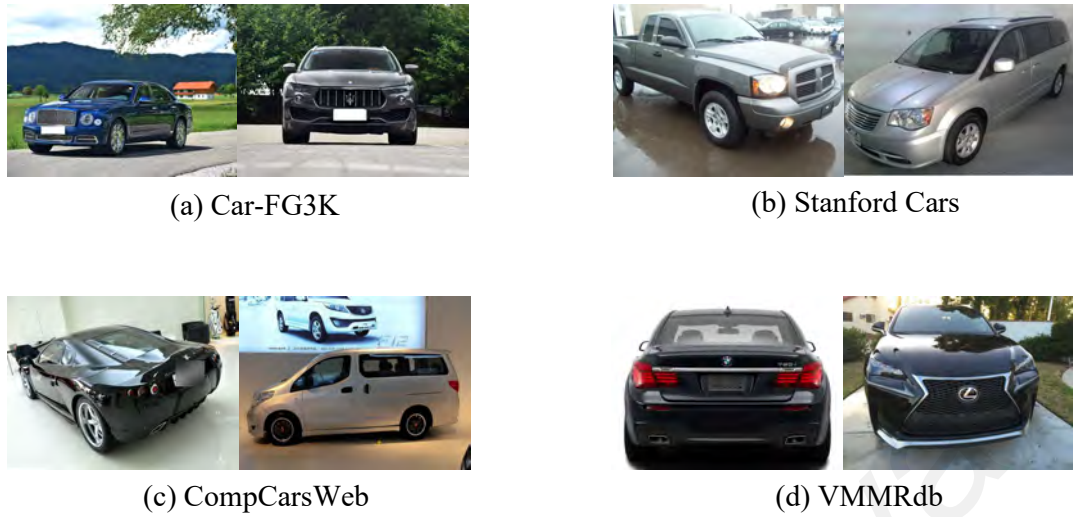


Figure 7.2: Sample Images from Car-FG3K, Stanford Cars, CompCarsWeb and VMRRdb

Table 7.1: Datasets Statistics

	Car-FG3K	Stanford Cars	CompCarsWeb	VMRRdb
#Train	10,676	8,144	36,456	192,494
#Test	10,661	8,044	15,627	82,507
#Classes	1,892	196	431	497

7.4.2 Implementation Details

The experiment is conducted using the PyTorch framework. The training pipeline executes several data augmentation techniques to prevent overfitting and they include random cropping, random horizontal flipping and resizing. The training is performed for 90 epochs with an initial learning rate of 0.01. Stochastic Gradient Descent (SGD) with 0.9 momentum and $5e-4$ weight decay is chosen as the optimizer. For Car-FG3K (Wu et al., 2022) and VMRRdb (Tafazzoli et al., 2017), a cosine learning rate scheduler is used to adjust the learning rate after every training iteration. For Stanford Cars (Krause et al., 2013) and CompCarsWeb (Yang et al., 2015), the step learning rate scheduler is chosen where the learning rate is decayed by a factor of 10 for every 40 epochs. Unless stated otherwise, TRResNet-L (Ridnik et al., 2021a) is used as the backbone network to construct AGNet. No bounding box information is used throughout the experiment.

7.5 Results & Discussions

7.5.1 Quantitative Analysis

Table 7.2 portrays the performance of the state-of-the-art networks on Car-FG3K (Wu et al., 2022). Since the performances of these networks are not benchmarked against Car-FG3K in the original works, the training based on the implementation details unraveled in Chapter 7.4.2 is performed. Based on 224×224 resolution, the proposed network, AGNet ($N_{FE} = 3, g = 2, \mathcal{E} = 0.1, \alpha_{center} = 1e - 4$) reports 87.8% accuracy. As compared to the baseline, there is a 2.7% improvement margin. It is conjectured that such prominent improvement is due to the AG module that amalgamates the scale-varying information from various pyramid levels. The resultant cross-granularity features enable the network to pivot on both dense abstract information and granular visual cues in rendering an accurate judgment for VMMR.

Table 7.2: Performance Benchmarking on Car-FG3K

Reference	Backbone	Accuracy
TResNet-L (Ridnik et al., 2021a)	-	85.1%
TransFG (He et al., 2021)	ViT_B_16	78.2%
APCNN (Ding et al., 2021)	ResNet50	82.3%
APINet (Zhuang et al., 2020)	ResNet50	83.5%
ConvNeXt-S (Zhuang Liu et al., 2022)	-	84.1%
SwinV2-S (Ze Liu et al., 2022)	-	84.5%
MMALNet (F. Zhang et al., 2021)	ResNet50	84.9%
LRAU (Boukerche & Ma, 2021)	ResNet50	84.9%
Cross-X Learning (Luo et al., 2019)	ResNet50	85.0%
PMG (Du et al., 2020)	ResNet50	85.3%
AGNet	TResNet-L	87.8%

The Fine-Grained Transformer (TransFG) (He et al., 2021) is a transformer-based network designed for fine-grained classification tasks. It features the Part Selection Module (PSM) to attenuate inconsequential information by feeding only the discriminative image tokens to the classification head. Under the low data regime, the lack of inductive bias disadvantage is eminent where it reports 78.2% accuracy. Attention

Pyramid (AP) CNN (Ding et al., 2021) generates multi-scale features through FPN (Lin et al., 2017a) and AP. Its performance is less favorable as compared to AGNet since it pays attention to the immediate previous layer and the cross-layer information fails to transmit to a common end for aggregation purposes. APINet (Zhuang et al., 2020) outperforms APCNN and achieves 83.5% accuracy by capturing contrastive clues through the pairwise interaction of an image pair. Having around 50M parameters, ConvNeXt-S (Zhuang Liu et al., 2022) and SwinV2-S (Ze Liu et al., 2022) which are similar in size to TResNet-L also deliver stunning performance. Nevertheless, without leveraging the shallow-layer features, their performances lag AGNet by an average of 3.5%. Multi-Branch and Multi-Scale Learning Network (MMALNet) (F. Zhang et al., 2021) (84.9%) is a part-based network that localizes the discriminative parts by examining the activation values on the feature maps. The localization technique is time-consuming since repetitive forward passes are required. Lightweight RAU (LRAU) (Boukerche & Ma, 2021) is an enhanced version of RAU (Ma & Boukerche, 2020) in terms of classification performance and parameter efficiency. It refines the attention state iteratively based on the attention mask information. Due to the utilization of 1×1 , stride 2 convolution as downsampling operation, there is an inevitable loss of information and this compromises the feature distinctiveness. It delivers 84.9% accuracy. In Cross-X Learning (Luo et al., 2019), One-Squeeze Multi-Excitation is utilized to generate multi-scale features. In particular, the scale-specific features are recalibrated along the channel axis based on information saliency before being merged. Similar to APCNN, paying attention to the information at adjacent scales refrains the learning process of the network and 85.0% accuracy is reported. PMG (Du et al., 2020) reports 85.3% accuracy. It mines the discriminative features through the jigsaw puzzle strategy and the prediction is determined by the classification logits from multiple pyramid levels. However, PMG is

subpar as compared to AGNet since it treats the features at their respective granularities without exploiting the combined benefit of coarse- and fine-grained information.

Table 7.3 illustrates the network performances benchmarked on Stanford Cars (Krause et al., 2013). For a fair comparison with other works, the bounding box information is utilized for 224×224 resolutions. The evaluation has seen AGNet ($N_{FE} = 3, g = 4, \mathcal{E} = 0.1, \alpha_{center} = 1e - 4$) advances the performance of the baseline network by 1.1% to 94.8%. Spatially Weighted Pooling (SWP) (Hu et al., 2017) adopts a learnable pooling layer to provide better summarization of the feature information according to spatial importance. However, the performance of SWP is suboptimal since it is translation intolerant, especially under multi-view circumstances. ConvAM (Yu et al., 2020) reports 93.1% but it is bulky and does not suit lightweight devices. Cross Attention-Multi-scale Sparse Network (CA-MSNet) (Maopeng Li et al., 2022) which puts forward a cross-attention mechanism module and a multi-scale sparse structure module for multi-scale features learning delivers 93.5% accuracy. Channel Max Pooling (CMP) (Ma et al., 2019) is a novel pooling strategy that retains the salient channel information but its performance (93.7%) is less competitive as compared to AGNet. The AGNet also outflanks RAU (Ma & Boukerche, 2020) and LRAU (Boukerche & Ma, 2021) by an average of 1%.

On 448×448 resolution, no bounding box information is utilized. The AGNet ($N_{FE} = 3, g = 4, \mathcal{E} = 0.1, \alpha_{center} = 1e - 4$) outperforms the rest of the networks by achieving 95.5%, a 1.3% improvement over the baseline. The transformer-based architecture, namely Feature Concentration Transformer Plus (TransIFC+) (H. Liu et al., 2023) and TransFG (He et al., 2021), are at least 0.7% lower than AGNet and their performances can be boosted if more training images are available. For Data Augmented Dual-Attention Interactive Network (DADAINet) (Zhu & Li, 2022), LSTM and MHSA are used concurrently to uncover fine-grained features and they report 94.9%. Comparing

MMALNet (F. Zhang et al., 2021) (95.0%) and PMG (Du et al., 2020) (95.1%), the latter consistently delivers better results on Car-FG3K and Stanford Cars and its performance is on par with EfficientNetv2-L (Tan & Le, 2021). Progressively Sampling Discriminative Parts Network (PSDPNet) (Guo et al., 2022) also acquires the same performance level owing to the ability to sample discriminative object parts. However, it requires the number of sampled parts to be set explicitly and this is counter-intuitive since different viewpoints may present different numbers of distinctive parts. APCNN (Ding et al., 2021), APINet (Zhuang et al., 2020) and Feature Covariance Attention (FCA) (Jung et al., 2023) are equally outstanding where they produce 95.3% accuracy, which is 0.2% lower than AGNet.

Table 7.3: Performance Benchmarking on Stanford Cars

Reference	Backbone	Input	BBox	Accuracy
TResNet-L (Ridnik et al., 2021a)	-	224 ²	✓	93.7%
SWP (Hu et al., 2017)	ResNet101	224 ²	✓	93.1%
ConvAM (Yu et al., 2020)	ResNet50	224 ²	✓	93.1%
CA-MSNet (Maopeng Li et al., 2022)	ResNet50	224 ²	✓	93.5%
RAU (Ma & Boukerche, 2020)	ResNet101	224 ²	✓	93.6%
CMP (Ma et al., 2019)	DenseNet161	224 ²	✓	93.7%
LRAU (Boukerche & Ma, 2021)	ResNet50	224 ²	✓	93.9%
AGNet	TResNet-L	224²	✓	94.8%
TResNet-L (Ridnik et al., 2021a)	-	448 ²	×	94.2%
PCB-MFF Network (L. Wang et al., 2022)	ResNet34	448 ²	×	93.4%
PyConv (G. Wang et al., 2021)	ResNet50	448 ²	×	93.6%
ViT (Dosovitskiy et al., 2020)	ViT_B_16	448 ²	×	93.7%
MLBPNNet (Ming Li et al., 2022)	ResNet34	448 ²	×	93.8%
EfficientNetv2-S (Tan & Le, 2021)	-	448 ²	×	93.8%
FIFFNet (P. Wang et al., 2022)	ResNet101	448 ²	×	94.1%
PLFENet (Lu et al., 2022)	ResNet101	448 ²	×	94.1%
GTCNet (Xiang et al., 2019)	DenseNet264	448 ²	×	94.3%
CLANet (Huang et al., 2022)	ResNet101	448 ²	×	94.5%
Cross-X Learning (Luo et al., 2019)	ResNet50	448 ²	×	94.6%
EfficientNetv2-M (Tan & Le, 2021)	-	448 ²	×	94.6%
EPCNN (Yu et al., 2022)	ResNet50	448 ²	×	94.6%
SA-MFNet (H. Chen et al., 2022)	-	448 ²	×	94.7%
TransIFC+ (H. Liu et al., 2023)	Swin-B	448	×	94.7%
TransFG (He et al., 2021)	ViT_B_16	448 ²	×	94.8%
DADAINet (Zhu & Li, 2022)	ResNet50	448 ²	×	94.9%
MMALNet (F. Zhang et al., 2021)	ResNet50	448 ²	×	95.0%
PMG (Du et al., 2020)	ResNet50	448 ²	×	95.1%

Table 7.3: Performance Benchmarking on Stanford Cars, continued

Reference	Backbone	Input	BBox	Accuracy
EfficientNetv2-L (Tan & Le, 2021)	-	448 ²	×	95.1%
PSDPNet (Guo et al., 2022)	ResNet50	448 ²	×	95.1%
APCNN (Ding et al., 2021)	ResNet50	448 ²	×	95.3%
APINet (Zhuang et al., 2020)	ResNet50	448 ²	×	95.3%
FCA (Jung et al., 2023)	ResNet50	448 ²	×	95.3%
AGNet	TResNet-L	448²	×	95.5%

Referring to Table 7.4, it is evident that TResNet-L (Ridnik et al., 2021a), being a strong backbone CNN, surpasses several networks that are designed for VMMR on CompCarsWeb (Yang et al., 2015) by reporting 97.8% accuracy. Although its performance seems to be saturated, upon incorporating the AG module, the classification accuracy is raised further by 0.8%, thus achieving 98.6% accuracy ($N_{FE} = 3, g = 2, \varepsilon = 0.1, \alpha_{center} = 1e - 4$). This result aligns with the expectation that the inclusion of both geometric and semantic information promotes the learning of discriminative and diverse feature representations. MAWNet (Ghassemi et al., 2019) also delivers decent performance i.e. 97.8% accuracy but the bounding box annotation is required at the pretraining stage of the classifier module. LRAU (Boukerche & Ma, 2021) renders better performance than RAU (Ma & Boukerche, 2020) and CMP (Ma et al., 2019) but its performance is 0.3% lower than AGNet. Global Topology Constraint Network (GTCNet) (Ma et al., 2019) is a network built on DenseNet264 (Huang et al., 2017) that enforces topology constraints through point-wise and depth-wise convolution. It produces 98.5% accuracy but the global topology relationship is considered for top-level feature maps only. Embedding Pose (EP) CNN (Yu et al., 2022) leverages the viewpoint information to enhance the VMMR performance. In particular, it injects the viewpoint embeddings extracted from Tiny-YOLOv3 (Redmon & Farhadi, 2018) into ResNet50 (He et al., 2016) to enrich the feature embeddings. Although the performance is on par with AGNet, it requires bounding box and viewpoints annotations and this eventually translates to an

expensive data preparation process. Moreover, its performance is 0.9% poorer than AGNet on Stanford Cars (Krause et al., 2013).

Table 7.4: Performance Benchmarking on CompCarsWeb

Reference	Backbone	Input	BBox	Accuracy
TResNet-L (Ridnik et al., 2021a)	-	224 ²	×	97.8%
ConvAM (Yu et al., 2020)	ResNet50	224 ²	×	95.3%
A3M (Han et al., 2018)	ResNet50	224 ²	×	95.4%
Co-occurrence Learning (Elkerdawy et al., 2018)	ResNet50	224 ²	×	95.6%
Fine-Tuning DARTS (Tanveer et al., 2021)	-	224 ²	×	95.9%
ViT (Dosovitskiy et al., 2020)	ViT_B_16	224 ²	×	96.2%
TransFG (He et al., 2021)	ViT_B_16	224 ²	×	96.7%
SWP (Hu et al., 2017)	ResNet101	224 ²	×	97.6%
PLFENet (Lu et al., 2022)	ResNet101	448 ²	×	97.7%
MAWNet (Ghassemi et al., 2019)	Wide ResNet50	224 ²	×	97.8%
RAU (Ma & Boukerche, 2020)	ResNet101	224 ²	✓	97.8%
CMP (Ma et al., 2019)	DenseNet161	224 ²	✓	97.9%
LRAU (Boukerche & Ma, 2021)	ResNet50	224 ²	✓	98.3%
GTCNet (Xiang et al., 2019)	DenseNet264	448 ²	×	98.5%
EPCNN (Yu et al., 2022)	ResNet50	224 ²	✓	98.6%
AGNet	TResNet-L	224²	×	98.6%

The performance of AGNet ($N_{FE} = 3, g = 2, \mathcal{E} = 0.1, \alpha_{center} = 1e - 4$) is validated further on the large-scale dataset, which is VMRRdb. Based on Table 7.5, the improvement brought by the AG module is encouraging. It improves the baseline by 0.6%, reporting 92.5% accuracy.

Table 7.5: Performance Benchmarking on VMRRdb

Reference	Backbone	Accuracy
TResNet-L (Ridnik et al., 2021a)	-	91.9%
AGNet	TResNet-L	92.5%

7.5.2 Qualitative Analysis

T-distributed Stochastic Neighbour Embedding (TSNE) (van der Maaten & Hinton, 2008) is applied to the feature embeddings of both the baseline (TResNet-L) and AGNet. In particular, the feature embeddings from the penultimate layer of both networks are

class are confined into a small feature space and this lowers the classification performance. On the contrary, adopting an optimal α_{center} for feature space regulation reduces the intraclass variance appropriately. Nevertheless, α_{center} is dataset-dependent and it should be identified empirically.

The most significant hyperparameter that affects the performance of the AG module is N_{FE} . It dictates the number of feature maps from early convolution layers that should be leveraged to generate multi-scale features. Based on experiment B, when the feature maps from the penultimate convolutional blocks are channeled into the AG module i.e. $N_{FE} = 2$, there is a formidable accuracy improvement from the baseline and it accounts for 1.9% and 1.1% for Car-FG3K and Stanford Cars, respectively. This manifests that the inclusion of mid-level feature maps augments the understanding of the vehicles further by complementing the high-level feature maps that are rich in global abstract information with exquisite vehicle details. When N_{FE} increases from 2 to 3, the increment in classification performance on Car-FG3K which accounts for 0.8% remains motivating. Sweeping beyond $N_{FE} = 3$, the performance does not progress further. This implies that over-involvement of local information does not strengthen the cross-granularity features since the information can be noisy and it may divert the network from picking up the discriminative details.

Experiment C investigates an appropriate g for GFConv during the FE stage to keep a balanced trade-off between the classification accuracy and computational complexity. In addition to parameter saving, grouped convolution promotes diversified learning by learning less correlated features across the groups. The experiment indicates that a larger network size does not guarantee better performance where $g = 2$ brings better performance than $g = 1$ on both datasets since the risk of overfitting is lowered. AGNet acquires the best performance i.e. 95.5% when $g = 4$ on Stanford Cars. When $g = 8$, the

performance of AGNet dips slightly on both datasets. It signifies that adopting a value larger than appropriate restricts the information flow across the channels and this abstains from the derivation of discriminative features (Xie et al., 2020).

Table 7.6: Ablation Study on Car-FG3K

Exp.	α_{Center}	N_{FE}	g	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	#Params (M)	GFLOPs
Baseline	0	-	-	85.1	85.7	82.4	82.3	48.0	8.8
A	1e-2	3	2	86.0	86.9	83.6	83.4	61.9	9.8
	1e-3	3	2	87.1	88.5	84.7	85.0	61.9	9.8
	1e-4	3	2	87.8	89.5	85.4	85.8	61.9	9.8
	1e-5	3	2	87.6	89.2	85.3	85.7	61.9	9.8
B	1e-4	2	2	87.0	88.3	84.7	84.9	54.6	9.3
	1e-4	3	2	87.8	89.5	85.4	85.8	61.9	9.8
	1e-4	4	2	87.5	88.8	85.3	85.6	69.5	10.6
C	1e-4	3	1	87.6	89.3	85.5	85.9	72.5	10.3
	1e-4	3	2	87.8	89.5	85.4	85.8	61.9	9.8
	1e-4	3	4	87.3	88.7	85.0	85.2	56.5	9.6
	1e-4	3	8	87.5	88.6	84.9	85.2	53.9	9.4

Table 7.7: Ablation Study on Stanford Cars

Exp.	α_{Center}	N_{FE}	g	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	#Params (M)	GFLOPs
Baseline	0	-	-	94.2	94.2	94.1	94.1	44.5	35.2
A	1e-2	3	2	89.1	90.0	89.0	89.2	59.1	39.2
	1e-3	3	2	95.3	95.3	95.2	95.1	59.1	39.2
	1e-4	3	2	95.4	95.4	95.4	95.3	59.1	39.2
	1e-5	3	2	95.4	95.4	95.3	95.3	59.1	39.2
B	1e-4	2	2	95.3	95.3	95.2	95.2	52.7	37.2
	1e-4	3	2	95.4	95.4	95.4	95.3	59.1	39.2
	1e-4	4	2	95.4	95.4	95.3	95.3	65.9	42.2
C	1e-4	3	1	95.3	95.2	95.2	95.2	69.8	41.3
	1e-4	3	2	95.4	95.4	95.4	95.3	59.1	39.2
	1e-4	3	4	95.5	95.5	95.3	95.3	53.8	38.2
	1e-4	3	8	95.3	95.4	95.3	95.2	51.2	37.7

7.5.4 Generalization Study

In this section, the generalization ability of the proposed AG module on other well-known CNNs i.e. VGG16 (Simonyan & Zisserman, 2014), Inceptionv3 (Szegedy et al., 2016), ResNet50 (He et al., 2016) and TResNet-L (Ridnik et al., 2021a) is examined.

Following the implementation details in Chapter 7.4.2, the training process is performed and the results are tabulated in Figure 7.4. The AG module settings for all networks are $N_{FE} = 3, g = 2, \mathcal{E} = 0.1, \alpha_{center} = 1e - 4$.

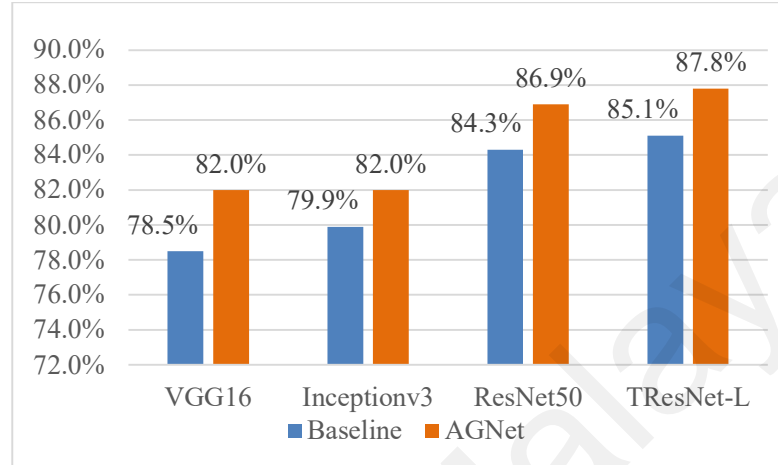


Figure 7.4: Compatibility between AG Module and Existing CNNs on Car-FCG3K

Among all the networks, the performance of TResNet-L-AG (87.8%) is the most astounding, followed by ResNet50-AG (86.9%), VGG16-AG (82.0%) and Inceptionv3-AG (82.0%). In terms of performance gain, VGG16-AG tops the list with an improvement margin as high as 3.5%. Given its shallow structure, the feature maps are still weak semantically. The AG module elevates the semanticity by putting the feature maps through the FE stage to make them more expressive. Subsequently, the FI stage performs information stitching from all scale levels to land a more accurate prediction. The improvements recorded by TResNet-L-AG and ResNet50-AG are 2.7% and 2.6%, respectively. The performance gain of Inceptionv3-AG which records 2.1% is also promising.

In terms of the number of parameters, the increment is the least for VGG16, which accounts for 1.7M only. This is because the channel count of the top-level feature maps stands at 512 only. For Inceptionv3, ResNet50 and TResNet-L, the increment

approximates 14M parameters and the resultant networks (Inceptionv3-AG: 39.3M, ResNet50-AG: 41.2M and TResNet-L-AG: 61.9M) are still moderate in size. To pursue lightweight networks, as portrayed in Chapter 7.5.3, g can be tuned to reduce the network size further with minimal loss in accuracy.

7.6 Conclusion

In this work, the AG module that leverages cross-granularity features generated from image pyramids for an accurate VMMR system is proposed. Most of the prevailing VMMR approaches ingest the top-level feature maps to infer the vehicle model. However, this piece of information is insufficient since the deep-layer features do not encompass the minute visual traits that are embedded in the shallow-layer features. The AG module addresses the drawback by establishing lateral connections to ingest the scale-specific components from early, mid and final convolution layers. The feature expressiveness is subsequently reinforced by the GFConv during the FE stage. In the FI stage, the refined scale-specific representations are merged in a nonlinear manner to produce multi-scale features that are enclosed with both fine-grained details and global abstract attributes of various vehicle models. During the experiment, it is proven that the AG module empowers the backbone networks by achieving state-of-the-art performances on Car-FG3K (87.8%), Stanford Cars (95.5%), CompCarsWeb (98.6%) and VMMRdb (92.5%) with improvement margin as high as 2.7%. In addition, the qualitative analysis reveals AGNet's exceptional learning capability as compared to the baseline where the learned feature embeddings are more cohesive. The generalization ability of the AG module is also demonstrated through integration with VGG16, ResNet, Inceptionv3 and TResNet-L with an average accuracy improvement of 2.7%.

CHAPTER 8: CONCLUSION

8.1 Conclusion

In this work, a total of 5 frameworks that advance the performance of vehicle recognition are deliberated. The Principal Component Analysis-Linear Discriminant Analysis-Convolutional Neural Network (PCA-LDA-CNN) has seen a hybrid approach of unsupervised and supervised dimension reduction techniques, namely PCA and LDA, as an alternative to the computationally expensive backpropagation technique in learning the convolution filters. A parameter-free Channel-Based Attention Module (ChBAM) is also introduced as an additional component in PCA-LDA-CNN for channel reweighting purposes. The proposed framework attains 99.6% and 97.8% accuracies on Vehicle Make and Model Recognition (VMMR) datasets with 30 and 300 classes, respectively. Furthermore, the ablation study reveals the efficacy of the methodology where PCA-LDA-CNN outperforms the backpropagation-optimized CNN by 0.6% whereas ChBAM contributes about 0.2% towards the classification performance. A thorough robustness test that delves into the vulnerability of PCA-LDA-CNN against various environmental-induced distortions is conducted and it is affirmed that the network is highly versatile.

To elevate the differentiation ability of the CNN, a Spatial Attention Module (SAM) is devised. Powered by Multi-Head Self-Attention (MHSA), it harnesses the global relevancy of all spatial positions from the top-level feature maps to underscore the area that carries exclusive vehicle information. SAM raises the performance of CaffeNet by 1.3% on the Beijing Institute of Technology (BIT)-Vehicle dataset. The highest performance is reported by ResNet50-SAM where it outperforms works originating from the attention domain with 98.2% accuracy. A dissection of the feature embeddings learned by SAM manifests its high sensitivity to the uniqueness of various vehicle types. Additionally, the design of SAM is rationalized under the ablation study and the

generalization study demonstrates the high compatibility between SAM and existing CNNs with an average 0.7% improvement margin.

The cognizant ability of the top-level feature maps, especially for VMMR tasks, is hampered due to the lack of fine-grained microscopic details. A Cross-Granularity (CG) module is tasked to compile the feature maps from various pyramid levels in one shot and subsequently synthesize multi-scale feature maps through dilated convolutions and 1×1 convolutions. The ResNet50-based CG Network (CGNet) achieves 98.6%, 95.4%, 86.4% and 99.1% accuracies on Web-Nature Comprehensive Cars (CompCarsWeb), Stanford Cars, Car-FG3K and Surveillance-Nature Comprehensive Cars (CompCarsSV) datasets, respectively with an improvement margin as high as 2.2% from the baseline. The qualitative analysis also demonstrates the ability of the CG module to pinpoint the extraordinary vehicle parts instead of the general region to maximize the discrimination capability. Moreover, integrating the CG module into other CNN backbones such as Inceptionv3 and DenseNet169 asserts the generalization ability by recording an average improvement of 0.9% accuracy.

To maintain a balance between network performance and computational complexity in multi-scale feature learning, a Coarse-to-Fine Context Aggregation (CFCA) module is designed. It implements multi-stream convolutions that run on different dilation rates to expand the field of view of the convolution kernels. In addition, grouped convolution is employed to ensure a modest network size without sacrificing the quality of the representation learning process. In essence, the CFCA module distills information from two scale levels at a time in a recursive manner to eventually form multi-granularity feature maps that are rich in both large-scale and small-scale components for a better comprehension of the vehicle model information. The experimental results demonstrate the exceptional performance raise which ranges from 0.9% to 1.9% accuracy on 5

publicly available VMMR datasets upon incorporating the CFCA module. The visualization of learned feature maps indicates the high inclusiveness and strong cohesiveness of the CFCA Network (CFCANet) as compared to the baseline. More importantly, the CFCA module is modular and handy to use with other CNNs where 1.1% and 1.8% accuracy gains are declared based on CompCarsWeb and Stanford Cars datasets, respectively.

In pursuit of a performance-oriented VMMR solution, an Augmented-Granularity (AG) module that empowers the AG Network (AGNet) is featured. The AG module implements grouped focus convolution (GFConv) to enhance the information retention and distillation of scale-specific components before synthesizing multi-scale feature maps. The proposed framework, TResNet-L-based AGNet, acquires 87.8%, 95.5%, 98.6% and 92.5% accuracies on Car-FG3K, Stanford Cars, CompCarsWeb and VMMRdb datasets, respectively. Furthermore, visual inspection of the feature embeddings reveals the ability of the AG module to learn a more compact feature representation. The generalization study further implies the high modularity characteristics of the AG module where it brings substantial accuracy improvement that goes as high as 3.5% on a variety of CNNs.

8.2 Future Works

To reach an impeccable state for vehicle recognition solutions, there are still works that remain to be done. At a glance, they consist of solution robustness, model relevancy, multi-modality and emerging architectures. The suggested future works are listed as follows:

- **Solution robustness.** To further improve the robustness of the solution, it is essential to enrich the vehicle datasets further. Although the existing datasets are highly diversified, the images captured under extreme weather conditions such as

heavy rain, fog and snow are barely seen. Having not considered these occurrences during training will bring significant performance drift. Therefore, it is essential to incorporate the aforementioned scenarios into the datasets to reinforce the generalization ability of the solution.

- **Model relevancy.** Since the vehicle recognition solutions are trained to recognize known vehicle classes, they are bound to fail when an unseen vehicle appears. This issue is a huge stumbling block to the sustainability of the deployed solution. As time progresses, they may cease to be relevant since the legacy vehicle models are eliminated whereas new vehicle models are introduced into the market. The most primitive succor to this is conducting model retraining but this is an expensive and time-consuming process. Therefore, it is meaningful to consider continual learning strategies that enable the model to acquire new information whilst retaining the existing knowledge. In addition, few-shot learning that adapts the model based on a limited number of samples can be exploited to reduce the training-to-deployment time significantly. Most important of all, an automated pipeline needs to be implemented so that the entire training and deployment pipeline can run seamlessly without human intervention.
- **Multi-modality.** Most of the vehicle recognition solutions are designed to be homogeneous systems. A homogeneous system is easier to maintain but it may be less reliable. For instance, severe object occlusion will incapacitate the Computer-Vision (CV)-based approaches and hardware failure will lead to system downtime. To increase the system reliability, a multi-modal approach, which encompasses sensors and CV, can be studied. The resultant solution will harness the strengths and weaknesses of individual components and thereby further improve vehicle recognition performance by learning more diverse and richer feature embeddings.

- **Emerging architectures.** Deep learning architectures have continued pushing the boundary of various CV tasks including vehicle recognition. Starting with CNN, convolution derives hierarchical features by enforcing local connectivity among the features within the convolution kernels. As a unified architecture for Natural Language Processing (NLP), CV, and audio, transformer architectures have also become the focus of the research community. Nonetheless, none of them are impeccable since CNN fails to track long-range dependencies whereas the transformer has quadratic complexity. Moving forward, these flaws can be fixed by improvising the CNN architectures (Ridnik et al., 2021a; Tan & Le, 2021) or eliminating the complexity of MHSA (Jaegle et al., 2021; Jeevan & Sethi, 2021). In addition, marrying the merits of both architectures (Dai et al., 2021) is worth exploring. Another line of research can be the automated optimization of deep learning architectures through Neural Architecture Search (NAS) (Elsken et al., 2019). NAS successfully eliminates biased decisions and laborious trial and error processes during the network design stage. Under the umbrella of NAS, one may investigate different approaches such as reinforcement learning, evolutionary algorithms, and Bayesian optimization to deduce novel competitive architectures.

REFERENCES

- Adarsh, P., Rathi, P., & Kumar, M. (2020). YOLO v3-Tiny: Object Detection and Recognition using one stage improved model. 2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS).
- Ahn, S., Kandala, S., Uzan, J., & El-Basyouny, M. (2011). Impact of traffic data on the pavement distress predictions using the mechanistic empirical pavement design guide. *Road Materials and Pavement Design*, 12(1), 195-216.
- Alghamdi, A. S., Saeed, A., Kamran, M., Mursi, K. T., & Almukadi, W. S. (2023). Vehicle Classification Using Deep Feature Fusion and Genetic Algorithms. *Electronics*, 12(2), 280.
- Ali, M., Tahir, M. A., & Durrani, M. N. (2022). Vehicle images dataset for make and model recognition. *Data in Brief*, 42.
- Amirkhani, A., & Barshooi, A. H. (2022). DeepCar 5.0: Vehicle Make and Model Recognition Under Challenging Conditions. *IEEE Transactions on Intelligent Transportation Systems*.
- Arandjelović, R., & Zisserman, A. (2012). Three things everyone should know to improve object retrieval. 2012 IEEE Conference on Computer Vision and Pattern Recognition.
- Arinaldi, A., Pradana, J. A., & Gurusinga, A. A. (2018). Detection and classification of vehicles for traffic video analytics. *Procedia Computer Science*, 144, 259-268.
- Asborno, M. I., Burris, C. G., & Hernandez, S. (2019). Truck body-type classification using single-beam LiDAR sensors. *Transportation Research Record*, 2673(1), 26-40.
- Bai, S., Liu, Z., & Yao, C. (2018). Classify vehicles in traffic scene images with deformable part-based models. *Machine Vision and Applications*, 29(3), 393-403.
- Balid, W., Tafish, H., & Refai, H. H. (2017). Intelligent vehicle counting and classification sensor for real-time traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems*, 19(6), 1784-1794.
- Baruah, J. K., Kumar, A., Bera, R., & Dhar, S. (2019). Autonomous Vehicle—A Miniaturized Prototype Development. In *Advances in Communication, Devices and Networking* (pp. 317-324). Springer.

- Baser, E., & Altun, Y. (2016). Detection and classification of vehicles in traffic by using haar cascade classifier. *Proc. 58th ISERD Int. Conf. Sci. Innov. Eng.*, no. December.
- Basheer Ahmed, M. I., Zaghdoud, R., Ahmed, M. S., Sendi, R., Alsharif, S., Alabdulkarim, J., . . . Krishnasamy, G. (2023). A real-time computer vision based approach to detection and classification of traffic incidents. *Big Data and Cognitive Computing*, 7(1), 22.
- Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3), 346-359.
- Behera, A., Wharton, Z., Hewage, P. R., & Bera, A. (2021). Context-aware attentional pooling (cap) for fine-grained visual classification. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Bello, I., Zoph, B., Vaswani, A., Shlens, J., & Le, Q. V. (2019). Attention augmented convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision*.
- Bernas, M., Płaczek, B., Korski, W., Loska, P., Smyła, J., & Szymała, P. (2018). A survey and comparison of low-cost sensing technologies for road traffic monitoring. *Sensors*, 18(10), 3243.
- Besbes, M. D., Kessentini, Y., & Tabia, H. (2020). Multi-stream Deep Networks for Vehicle Make and Model Recognition. *VISIGRAPP (5: VISAPP)*.
- Biglari, M., Soleimani, A., & Hassanpour, H. (2017a). A cascaded part-based system for fine-grained vehicle classification. *IEEE Transactions on Intelligent Transportation Systems*, 19(1), 273-283.
- Biglari, M., Soleimani, A., & Hassanpour, H. (2017b). Part-based recognition of vehicle make and model. *IET Image Processing*, 11(7), 483-491.
- Bischof, H., Godec, M., Leistner, C., Rinner, B., & Starzacher, A. (2010). Autonomous audio-supported learning of visual classifiers for traffic monitoring. *IEEE Intelligent Systems*, 25(3), 15-23.
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Boonsirisumpun, N., & Surinta, O. (2022). Fast and Accurate Deep Learning Architecture on Vehicle Type Recognition. *Current Applied Science and Technology*, 16 pages-16 pages.

- Bortoluzzi, E. M., Schmidt, P. H., Brown, R. E., Jensen, M., Mancke, M. R., Larson, R. L., . . . White, B. J. (2023). Image Classification and Automated Machine Learning to Classify Lung Pathologies in Deceased Feedlot Cattle. *Veterinary Sciences*, *10*(2), 113.
- Bottero, M., Dalla Chiara, B., & Deflorio, F. P. (2013). Wireless sensor networks for traffic monitoring in a logistic centre. *Transportation Research Part C: Emerging Technologies*, *26*, 99-124.
- Boukerche, A., & Ma, X. (2021). A Novel Smart Lightweight Visual Attention Model for Fine-Grained Vehicle Recognition. *IEEE Transactions on Intelligent Transportation Systems*.
- Boukerche, A., Siddiqui, A. J., & Mammeri, A. (2017). Automated vehicle detection and classification: Models, methods, and techniques. *ACM Computing Surveys (CSUR)*, *50*(5), 1-39.
- Cao, F., Chen, S., Zhong, J., & Gao, Y. (2023). Traffic Condition Classification Model Based on Traffic-Net. *Computational Intelligence and Neuroscience*, 2023.
- Castello, P., Coelho, C., Del Ninno, E., Ottaviani, E., & Zanini, M. (1999). Traffic monitoring in motorways by real-time number plate recognition. *Proceedings 10th International Conference on Image Analysis and Processing*, 1128-1131.
- Chakraborti, T., McCane, B., Mills, S., & Pal, U. (2018). LOOP descriptor: local optimal-oriented pattern. *IEEE Signal Processing Letters*, *25*(5), 635-639.
- Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., & Ma, Y. (2015). PCANet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, *24*(12), 5017-5032.
- Chen, H., Cheng, L., Huang, G., Zhang, G., Lan, J., Yu, Z., . . . Ling, W.-K. (2022). Fine-grained visual classification with multi-scale features based on self-supervised attention filtering mechanism. *Applied Intelligence*, 1-17.
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European Conference on Computer Vision (ECCV)*.
- Chen, L. (2023). Road vehicle recognition algorithm in safety assistant driving based on artificial intelligence. *Soft Computing*, *27*(2), 1153-1162.

- Chen, S., Li, Z., Li, J., & Ye, H. (2022). Remote Sensing Image Classification of Sugarcane Harvest based on Tensorflow. 2022 9th International Conference on Digital Home (ICDH).
- Chen, S., Su, C., Kuang, Z., Ye, O., & Gong, X. (2020). Real-time detection of UAV detection image of power line insulator bursting based on YOLOV3. *Journal of Physics: Conference Series*.
- Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Cho, K., Van Merriënboer, B., Bahdanau, D., & Bengio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *SSST@EMNLP*.
- Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Chopra, S., Hadsell, R., & LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05).
- Chou, P.-Y., Kao, Y.-Y., & Lin, C.-H. (2023). Fine-grained Visual Classification with High-temperature Refinement and Background Suppression. *arXiv preprint arXiv:2303.06442*.
- Clady, X., Negri, P., Milgram, M., & Poulencard, R. (2008). Multi-class vehicle type recognition system. *IAPR Workshop on Artificial Neural Networks in Pattern Recognition*.
- Cline, D., Jeschke, S., White, K., Razdan, A., & Wonka, P. (2009). Dart throwing on surfaces. *Computer Graphics Forum*.
- Cyganek, B., & Woźniak, M. (2014). Vehicle logo recognition with an ensemble of classifiers. *Asian Conference on Intelligent Information and Database Systems*.
- Dai, Z., Liu, H., Le, Q. V., & Tan, M. (2021). Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34, 3965-3977.
- Datondji, S. R. E., Dupuis, Y., Subirats, P., & Vasseur, P. (2016). A survey of vision-based traffic monitoring of road intersections. *IEEE Transactions on Intelligent Transportation Systems*, 17(10), 2681-2698.

- Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition.
- Derrouz, H., Elbouziady, A., Abdelali, H. A., Thami, R. O. H., El Fkihi, S., & Bourzeix, F. (2019). Moroccan video intelligent transport system: Vehicle type classification based on three-dimensional and two-dimensional features. *IEEE Access*, 7, 72528-72537.
- Ding, Y., Ma, Z., Wen, S., Xie, J., Chang, D., Si, Z., . . . Ling, H. (2021). AP-CNN: Weakly supervised attention pyramid convolutional neural network for fine-grained visual classification. *IEEE Transactions on Image Processing*, 30, 2826-2836.
- Dong, H., Wang, X., Zhang, C., He, R., Jia, L., & Qin, Y. (2018). Improved robust vehicle detection and identification based on single magnetic sensor. *IEEE Access*, 6, 5247-5255.
- Dong, Z., Wu, Y., Pei, M., & Jia, Y. (2015). Vehicle type classification using a semisupervised convolutional neural network. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 2247-2256.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., . . . Gelly, S. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations.
- Du, R., Chang, D., Bhunia, A. K., Xie, J., Ma, Z., Song, Y.-Z., & Guo, J. (2020). Fine-grained visual classification via progressive multi-granularity training of jigsaw patches. European Conference on Computer Vision.
- Du, Y., Rui, T., Li, H., Yang, C., & Wang, D. (2023). DeepBP: A bilinear model integrating multi-order statistics for fine-grained recognition. *Computers and Electrical Engineering*, 105, 108432.
- Dubská, M., Herout, A., & Sochor, J. (2014). Automatic Camera Calibration for Traffic Understanding. BMVC.
- Elkerdawy, S., Ray, N., & Zhang, H. (2018). Fine-grained vehicle classification with unsupervised parts co-occurrence learning. Proceedings of the European Conference on Computer Vision (ECCV) Workshops.
- Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. *The Journal of Machine Learning Research*, 20(1), 1997-2017.

- Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2), 303-338.
- Fang, J., Zhou, Y., Yu, Y., & Du, S. (2016). Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture. *IEEE Transactions on Intelligent Transportation Systems*, 18(7), 1782-1792.
- Farady, I., Kuo, C.-C., Ng, H.-F., & Lin, C.-Y. (2023). Hierarchical Image Transformation and Multi-Level Features for Anomaly Defect Detection. *Sensors*, 23(2), 988.
- Forcen, J. I., Pagola, M., Barrenechea, E., & Bustince, H. (2020). Co-occurrence of deep convolutional features for image search. *Image and Vision Computing*, 97, 103909.
- Fu, J., Zheng, H., & Mei, T. (2017). Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Gao, H., Miao, Y., Cao, X., & Li, C. (2021). Densely connected multiscale attention network for hyperspectral image classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 2563-2576.
- Gao, Y., Han, X., Wang, X., Huang, W., & Scott, M. (2020). Channel interaction networks for fine-grained image categorization. Proceedings of the AAAI Conference on Artificial Intelligence.
- Gao, Y., & Lee, H. J. (2016). Local tiled deep networks for recognition of vehicle make and model. *Sensors*, 16(2), 226.
- Ge, Y., Hu, J., & Deng, W. (2017). PCA-LDANet: A Simple Feature Learning Method for Image Classification. *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, 370-375.
- Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. International Conference on Machine Learning.
- Ghassemi, S., Fiandrotti, A., Caimotti, E., Francini, G., & Magli, E. (2019). Vehicle joint make and model recognition with multiscale attention windows. *Signal Processing: Image Communication*, 72, 69-79.

- Gholamalinejad, H., & Khosravi, H. (2021a). IRVD: A Large-Scale Dataset for Classification of Iranian Vehicles in Urban Streets. *Journal of AI and Data Mining*, 9(1), 1-9.
- Gholamalinejad, H., & Khosravi, H. (2021b). Vehicle Classification using a Real-Time Convolutional Structure based on DWT pooling layer and SE blocks. *Expert Systems with Applications*, 115420.
- Gholamhosseinian, A., & Seitz, J. (2021). Vehicle Classification in Intelligent Transport Systems: An Overview, Methods and Software Perspective. *IEEE Open Journal of Intelligent Transportation Systems*.
- Girshick, R. (2015). Fast r-cnn. Proceedings of the IEEE International Conference on Computer Vision.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jégou, H., & Douze, M. (2021). LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference. Proceedings of the IEEE/CVF International Conference on Computer Vision.
- Grijalva, I., Spiesman, B. J., & McCornack, B. (2023). Image classification of sugarcane aphid density using deep convolutional neural networks. *Smart Agricultural Technology*, 3, 100089.
- Guan, J., Fei, J., Li, W., Jiang, X., Wu, L., Liu, Y., & Xi, J. (2023). Defect classification for specular surfaces based on deflectometry and multi-modal fusion network. *Optics and Lasers in Engineering*, 163, 107488.
- Gulzar, Y. (2023). Fruit Image Classification Model Based on MobileNetV2 with Deep Transfer Learning Technique. *Sustainability*, 15(3), 1906.
- Guo, C., Lin, Y., Chen, S., Zeng, Z., Shao, M., & Li, S. (2022). From the whole to detail: Progressively sampling discriminative parts for fine-grained recognition. *Knowledge-Based Systems*, 235, 107651.
- Haferkamp, M., Al-Askary, M., Dorn, D., Sliwa, B., Habel, L., Schreckenberger, M., & Wietfeld, C. (2017). Radio-based traffic flow detection and vehicle classification for future smart cities. 2017 IEEE 85th Vehicular Technology Conference (VTC Spring).

- Hamaguchi, R., Fujita, A., Nemoto, K., Imaizumi, T., & Hikosaka, S. (2018). Effective use of dilated convolutions for segmenting small object instances in remote sensing imagery. 2018 IEEE Winter Conference on Applications of Computer Vision (WACV).
- Han, K., Guo, J., Zhang, C., & Zhu, M. (2018). Attribute-aware attention model for fine-grained representation learning. Proceedings of the 26th ACM International Conference on Multimedia.
- Hanselmann, H., & Ney, H. (2020a). Elope: Fine-grained visual classification with efficient localization, pooling and embedding. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.
- Hanselmann, H., & Ney, H. (2020b). Fine-grained visual classification with efficient end-to-end localization. *arXiv preprint arXiv:2005.05123*.
- Hao, F., Liu, X., Li, M., & Han, W. (2023). Accurate Kidney Pathological Image Classification Method Based on Deep Learning and Multi-Modal Fusion Method with Application to Membranous Nephropathy. *Life*, 13(2), 399.
- He, H., Shao, Z., & Tan, J. (2015). Recognition of car makes and models from a single traffic-camera image. *IEEE Transactions on Intelligent Transportation Systems*, 16(6), 3182-3192.
- He, J., Chen, J.-N., Liu, S., Kortylewski, A., Yang, C., Bai, Y., . . . Yuille, A. (2021). TransFG: A Transformer Architecture for Fine-grained Recognition. AAAI Conference on Artificial Intelligence.
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Hedeya, M. A., Eid, A. H., & Abdel-Kader, R. F. (2020). A super-learner ensemble of deep networks for vehicle-type classification. *IEEE Access*, 8, 98266-98280.
- Hendrycks, D., & Gimpel, K. (2016). Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

- Hermann, D. S. (2018). Automotive displays-trends, opportunities and challenges. 2018 25th International Workshop on Active-Matrix Flatpanel Displays and Devices (AM-FPD).
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- Horn, G. V., Mac Aodha, O., Song, Y., Cui, Y., Sun, C., Shepard, A., . . . Belongie, S. (2018). The inaturalist species classification and detection dataset. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Hsieh, J.-W., Chen, L.-C., & Chen, D.-Y. (2014). Symmetrical SURF and its applications to vehicle detection and vehicle make and model recognition. *IEEE Transactions on Intelligent Transportation Systems*, 15(1), 6-20.
- Hu, B., Zhang, C., Wang, L., Zhang, Q., & Liu, Y. (2020). Multi-label x-ray imagery classification via bottom-up attention and meta fusion. Proceedings of the Asian Conference on Computer Vision.
- Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Hu, Q., Wang, H., Li, T., & Shen, C. (2017). Deep CNNs with spatially weighted pooling for fine-grained car recognition. *IEEE Transactions on Intelligent Transportation Systems*, 18(11), 3147-3156.
- Hu, T., Qi, H., Huang, Q., & Lu, Y. (2019). See better before looking closer: Weakly supervised data augmentation network for fine-grained visual classification. *arXiv preprint arXiv:1901.09891*.
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Huang, R., Wang, Y., & Yang, H. (2022). Cross-layer attention network for fine-grained visual categorization. *arXiv preprint arXiv:2210.08784*.
- Huang, Y., Wu, R., Sun, Y., Wang, W., & Ding, X. (2015). Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy. *IEEE Transactions on Intelligent Transportation Systems*, 16(4), 1951-1960.

- Hyvärinen, A., & Hoyer, P. O. (2001). Topographic independent component analysis as a model of V1 organization and receptive fields. *Neurocomputing*, 38, 1307-1315.
- Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.
- Ingle, S., & Phute, M. (2016). Tesla autopilot: semi autonomous driving, an uptick for future autonomy. *International Research Journal of Engineering and Technology*, 3(9), 369-372.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*.
- Ismael, A. M., & Şengür, A. (2021). Deep learning approaches for COVID-19 detection based on chest X-ray images. *Expert Systems with Applications*, 164, 114054.
- Jaderberg, M., Simonyan, K., & Zisserman, A. (2015). Spatial transformer networks. *Advances in Neural Information Processing Systems*, 28.
- Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., & Carreira, J. (2021). Perceiver: General Perception with Iterative Attention. *International Conference on Machine Learning*.
- Jain, N. K., Saini, R., & Mittal, P. (2019). A review on traffic monitoring system techniques. In *Soft Computing: Theories and Applications* (pp. 569-577). Springer.
- Jamil, A. A., Hussain, F., Yousaf, M. H., Butt, A. M., & Velastin, S. A. (2020). Vehicle Make and Model Recognition using Bag of Expressions. *Sensors*, 20(4), 1033.
- Jeevan, P., & Sethi, A. (2021). Vision Xformers: Efficient Attention for Image Classification. *arXiv preprint arXiv:2107.02239*.
- Jeng, S.-T., Chu, L., & Hernandez, S. (2013). Wavelet-k nearest neighbor vehicle classification approach with inductive loop signatures. *Transportation Research Record*, 2380(1), 72-80.
- Ji, R., Li, J., & Zhang, L. (2023). Siamese self-supervised learning for fine-grained visual classification. *Computer Vision and Image Understanding*, 229, 103658.

- Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., . . . Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. Proceedings of the 22nd ACM International Conference on Multimedia.
- Jung, H., Choi, M.-K., Jung, J., Lee, J.-H., Kwon, S., & Young Jung, W. (2017). ResNet-based vehicle classification and localization in traffic surveillance systems. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.
- Jung, Y., Syazwany, N. S., Kim, S., & Lee, S.-C. (2023). Fine-Grained Classification via Hierarchical Feature Covariance Attention Module. *IEEE Access*.
- Jyothi, P., Reddy, D. K., & Kumar, P. N. (2023). A Hybrid Classification Approach for Iris Recognition System for Security of Industrial Applications. *Journal of Scientific & Industrial Research*, 82(1), 151-157.
- Kang, H., Mo, S., & Shin, J. (2022). ReMixer: Object-aware Mixing Layer for Vision Transformers and Mixers. ICLR2022 Workshop on the Elements of Reasoning: Objects, Structure and Causality.
- Karthik, K., & Mahadevappa, M. (2023). Convolution neural networks for optical coherence tomography (OCT) image classification. *Biomedical Signal Processing and Control*, 79, 104176.
- Ke, X., Cai, Y., Chen, B., Liu, H., & Guo, W. (2023). Granularity-Aware Distillation and Structure Modeling Region Proposal Network for Fine-Grained Image Classification. *Pattern Recognition*, 109305.
- Ke, X., & Zhang, Y. (2020). Fine-grained vehicle type detection and recognition based on dense attention network. *Neurocomputing*, 399, 247-257.
- Khamayseh, Y., Mardini, W., & Tbashate, H. (2015). Leveraging The Data Gathering and Analysis Phases to Gain Situational Awareness. *Intelligent Automation & Soft Computing*, 21(4), 523-542.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. International Conference on Learning Representations.
- Krause, J., Gebru, T., Deng, J., Li, L.-J., & Fei-Fei, L. (2014). Learning features and parts for fine-grained recognition. 2014 22nd International Conference on Pattern Recognition.

- Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013). 3D Object Representations for Fine-Grained Categorization. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 554-561.
- Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*.
- Kul, S., Eken, S., & Sayar, A. (2017). A Concise Review on Vehicle Detection and Classification. 2017 International Conference on Engineering and Technology (ICET).
- Kumar, S., & Janet, B. (2022). DTMIC: Deep transfer learning for malware image classification. *Journal of Information Security and Applications*, 64, 103063.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541-551.
- Lee, H. J., Ullah, I., Wan, W., Gao, Y., & Fang, Z. (2019). Real-time vehicle make and model recognition with the residual SqueezeNet architecture. *Sensors*, 19(5), 982.
- Leotta, M. J., & Mundy, J. L. (2010). Vehicle surveillance with a generic, adaptive, 3d vehicle model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(7), 1457-1469.
- Li, A., Kang, B., Chen, J., Wu, D., & Zhou, L. (2022). Global Information-Assisted Fine-Grained Visual Categorization in Internet of Things. *IEEE Internet of Things Journal*, 10(1), 940-952.
- Li, A., Zhang, X., Li, P., & Kang, B. (2023). A teacher-student based attention network for fine-grained image recognition. *Digital Communications and Networks*.
- Li, B. (2010). Bayesian inference for vehicle speed and vehicle length using dual-loop detector data. *Transportation Research Part B: Methodological*, 44(1), 108-119.
- Li, G., Wu, G., Xu, G., Li, C., Zhu, Z., Ye, Y., & Zhang, H. (2023). Pathological image classification via embedded fusion mutual learning. *Biomedical Signal Processing and Control*, 79, 104181.

- Li, M., Lei, L., Sun, H., Li, X., & Kuang, G. (2022). Fine-grained visual classification via multilayer bilinear pooling with object localization. *The Visual Computer*, 38(3), 811-820.
- Li, M., Zhou, G., Cai, W., Li, J., Li, M., He, M., . . . Li, L. (2022). Multi-scale Sparse Network with Cross-Attention Mechanism for image-based butterflies fine-grained classification. *Applied Soft Computing*, 108419.
- Li, S., Lin, J., Li, G., Bai, T., Wang, H., & Pang, Y. (2018). Vehicle type detection based on deep learning in traffic scene. *Procedia Computer Science*, 131, 564-572.
- Li, Y., Song, B., Kang, X., Du, X., & Guizani, M. (2018). Vehicle-type detection based on compressed sensing and deep learning in vehicular networks. *Sensors*, 18(12), 4500.
- Liao, B., He, H., Du, Y., & Guan, S. (2022). Multi-component vehicle type recognition using adapted CNN by optimal transport. *Signal, Image and Video Processing*, 16(4), 975-982.
- Lin, S., He, Z., & Sun, L. (2023). A novel micro-defect classification system based on attention enhancement. *Journal of Intelligent Manufacturing*, 1-24.
- Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017a). Feature pyramid networks for object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017b). Focal loss for dense object detection. Proceedings of the IEEE International Conference on Computer Vision.
- Lin, Y.-L., Morariu, V. I., Hsu, W., & Davis, L. S. (2014). Jointly optimizing 3d model fitting and fine-grained classification. European Conference on Computer Vision.
- Liu, D., Wang, Y., Kato, J., & Mase, K. (2020). Contrastively-reinforced Attention Convolutional Neural Network for Fine-grained Image Recognition. BMVC.
- Liu, D., Zhao, L., Wang, Y., & Kato, J. (2023). Learn from each other to Classify better: Cross-layer mutual attention learning for fine-grained visual classification. *Pattern Recognition*, 140, 109550.
- Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. International Conference on Learning Representations.

- Liu, H., Tian, Y., Yang, Y., Pang, L., & Huang, T. (2016). Deep relative distance learning: Tell the difference between similar vehicles. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Liu, H., Zhang, C., Deng, Y., Xie, B., Liu, T., Zhang, Z., & Li, Y.-F. (2023). TransIFC: Invariant Cues-aware Feature Concentration Learning for Efficient Fine-grained Bird Image Classification. *IEEE Transactions on Multimedia*.
- Liu, M., Zhang, C., Bai, H., Zhang, R., & Zhao, Y. (2021). Cross-Part Learning for Fine-Grained Image Classification. *IEEE Transactions on Image Processing*, 31, 748-758.
- Liu, P., Fu, H., & Ma, H. (2021). An end-to-end convolutional network for joint detecting and denoising adversarial perturbations in vehicle classification. *Computational Visual Media*, 7(2), 217-227.
- Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. (2018). Path aggregation network for instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. *European Conference on Computer Vision*.
- Liu, W., & Zeng, K. (2018). SparseNet: A sparse DenseNet for image classification. *arXiv preprint arXiv:1804.05340*.
- Liu, X., Wang, L., & Han, X. (2022). Transformer with peak suppression and knowledge guidance for fine-grained image recognition. *Neurocomputing*, 492, 137-149.
- Liu, Z., Hu, H., Lin, Y., Yao, Z., Xie, Z., Wei, Y., . . . Dong, L. (2022). Swin transformer v2: Scaling up capacity and resolution. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., . . . Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF International Conference on Computer Vision*.
- Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 91-110.

- Lu, L., Wang, P., & Cao, Y. (2022). A novel part-level feature extraction method for fine-grained vehicle recognition. *Pattern Recognition*, 131, 108869.
- Luo, W., Yang, X., Mo, X., Lu, Y., Davis, L. S., Li, J., . . . Lim, S.-N. (2019). Cross-x learning for fine-grained visual categorization. Proceedings of the IEEE/CVF International Conference on Computer Vision.
- Luo, Z., Branchaud-Charron, F., Lemaire, C., Konrad, J., Li, S., Mishra, A., . . . Jodoin, P.-M. (2018). MIO-TCD: A new benchmark dataset for vehicle classification and localization. *IEEE Transactions on Image Processing*, 27(10), 5129-5141.
- Ma, X., & Boukerche, A. (2020). An AI-based visual attention model for vehicle make and model recognition. 2020 IEEE Symposium on Computers and Communications (ISCC).
- Ma, Z., Chang, D., & Li, X. (2019). Fine-Grained Vehicle Classification With Channel Max Pooling Modified CNNs. *IEEE Transactions on Vehicular Technology*, 68, 3224-3233.
- Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. Proc. ICML.
- Manzoor, M. A., & Morgan, Y. (2017). Vehicle Make and Model classification system using bag of SIFT features. 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC).
- Manzoor, M. A., Morgan, Y., & Bais, A. (2019). Real-time vehicle make and model recognition system. *Machine Learning and Knowledge Extraction*, 1(2), 611-629.
- Martín, A., & Tosunoglu, S. (2000). Image Processing Techniques for Machine Vision. Florida Conference on Recent Advances in Robotics.
- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5, 115-133.
- Mei, X., & Ling, H. (2011). Robust visual tracking and vehicle classification via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(11), 2259-2272.
- Meta, S., & Cinsdikici, M. G. (2010). Vehicle-classification algorithm based on component analysis for single-loop inductive detector. *IEEE Transactions on Vehicular Technology*, 59(6), 2795-2805.

- Murrugarra, R., Wallace, W., & Wojtowicz, J. (2010). *Task 30: Data Fusion Methodology*.
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. Proceedings of the 27th International Conference on Machine Learning (ICML-10).
- Nath, V., Chattopadhyay, C., & Desai, K. (2023). NSLNet: An improved deep learning model for steel surface defect classification utilizing small training datasets. *Manufacturing Letters*, 35, 39-42.
- Nazemi, A., Shafiee, M. J., Azimifar, Z., & Wong, A. (2020). Real-Time Vehicle Make and Model Recognition Using Unsupervised Feature Learning. *IEEE Transactions on Intelligent Transportation Systems*, 21, 3080-3090.
- Ngiam, J., Chen, Z., Bhaskar, S., Koh, P., & Ng, A. (2011). Sparse filtering. *Advances in Neural Information Processing Systems*, 24, 1125-1133.
- Ngiam, J., Chen, Z., Chia, D., Koh, P., Le, Q., & Ng, A. (2010). Tiled convolutional neural networks. *Advances in Neural Information Processing Systems*, 23, 1279-1287.
- Nordback, K., Kothuri, S., Phillips, T., Gorecki, C., & Figliozzi, M. (2016). Accuracy of bicycle counting with pneumatic tubes in Oregon. *Transportation Research Record*, 2593(1), 8-17.
- Odat, E., Shamma, J. S., & Claudel, C. (2017). Vehicle classification and speed estimation using combined passive infrared/ultrasonic sensors. *IEEE Transactions on Intelligent Transportation Systems*, 19(5), 1593-1606.
- Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2014). Learning and transferring mid-level image representations using convolutional neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Otto, C. W. (2006). Development of a mobile vehicle classification system.
- Pal, A., & Kumar, V. (2023). AgriDet: Plant Leaf Disease severity classification using agriculture detection framework. *Engineering Applications of Artificial Intelligence*, 119, 105754.
- Park, J., Woo, S., Lee, J.-Y., & Kweon, I. S. (2018). Bam: Bottleneck attention module. British Machine Vision Conference.

- Parsons, S. (2005). Ant Colony Optimization by Marco Dorigo and Thomas Stützle, MIT Press, 305 pp., \$40.00, ISBN 0-262-04219-3. *The Knowledge Engineering Review*, 20(1), 92-93.
- Paymode, A. S., & Malode, V. B. (2022). Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. *Artificial Intelligence in Agriculture*, 6, 23-33.
- Pearce, G., & Pears, N. (2011). Automatic make and model recognition from frontal images of cars. 2011 8th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS).
- Peng, F., Miao, Z., Li, F., & Li, Z. (2021). S-FPN: A shortcut feature pyramid network for sea cucumber detection in underwater images. *Expert Systems with Applications*, 182, 115306.
- Peng, Y., Jin, J. S., Luo, S., Xu, M., & Cui, Y. (2012). Vehicle type classification using PCA with self-clustering. 2012 IEEE International Conference on Multimedia and Expo Workshops.
- Perri, S., Spagnolo, F., Frustaci, F., & Corsonello, P. (2023). Welding defects classification through a Convolutional Neural Network. *Manufacturing Letters*, 35, 29-32.
- Petrovic, V. S., & Cootes, T. F. (2004). Analysis of Features for Rigid Structure Vehicle Type Recognition. BMVC.
- Phi, M. (2018). *Illustrated Guide to LSTM's and GRU's: A step by step explanation*. <https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21>
- Psyllos, A., Anagnostopoulos, C., & Kayafas, E. (2009). SIFT-based measurements for vehicle model recognition. Proceedings of the XIX IMEKO World Congress Fundamental and Applied Metrology, Lisbon, Portugal.
- Qibtiah, R. M., Zin, Z. M., & Hassan, M. F. A. (2023). Artificial intelligence system for driver distraction by stacked deep learning classification. *Bulletin of Electrical Engineering and Informatics*, 12(1), 365-372.
- Qin, H., Daquan, Z., Xu, T., Xie, E., Bian, Z., Cai, W., . . . Li, J. (2022). Defactorization Transformer: Modeling Long Range Dependency with Local Window Cost.

- Rachmadi, R. F., Uchimura, K., Koutaki, G., & Ogata, K. (2018). Single image vehicle classification using pseudo long short-term memory classifier. *Journal of Visual Communication and Image Representation*, 56, 265-274.
- Raja Abdullah, R. S. A., Abdul Aziz, N. H., Abdul Rashid, N. E., Ahmad Salah, A., & Hashim, F. (2016). Analysis on target detection and classification in LTE based passive forward scattering radar. *Sensors*, 16(10), 1607.
- Rajab, S., Al Kalaa, M. O., & Refai, H. (2016). Classification and speed estimation of vehicles via tire detection using single - element piezoelectric sensor. *Journal of Advanced Transportation*, 50(7), 1366-1385.
- Ramnath, K., Sinha, S. N., Szeliski, R., & Hsiao, E. (2014). Car make and model recognition using 3d curve alignment. IEEE Winter Conference on Applications of Computer Vision.
- Rao, Y., Chen, G., Lu, J., & Zhou, J. (2021a). Counterfactual attention learning for fine-grained visual categorization and re-identification. Proceedings of the IEEE/CVF International Conference on Computer Vision.
- Rao, Y., Zhao, W., Zhu, Z., Lu, J., & Zhou, J. (2021b). Global filter networks for image classification. *Advances in Neural Information Processing Systems*, 34.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*.
- Ridnik, T., Lawen, H., Noy, A., Ben Baruch, E., Sharir, G., & Friedman, I. (2021a). Tresnet: High performance gpu-dedicated architecture. Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision.
- Ridnik, T., Sharir, G., Ben-Cohen, A., Ben-Baruch, E., & Noy, A. (2021b). ML-Decoder: Scalable and Versatile Classification Head. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV).

- Roecker, M. N., Costa, Y. M., Almeida, J. L., & Matsushita, G. H. (2018). Automatic vehicle type classification with convolutional neural networks. 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP).
- Rong, W.-Z., Han, J., Cai, Y.-H., & Liu, G. (2021). Multi-Scale Discriminative Regions Attention Network for Fine-grained Vehicle Classification. *Journal of Network Intelligence*.
- Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.
- Samuel, A. L. (1959). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210-229.
- Sánchez, H. C., Parra, N. H., Alonso, I. P., Nebot, E., & Fernández-Llorca, D. (2021). Are We Ready for Accurate and Unbiased Fine-Grained Vehicle Classification in Realistic Environments? *IEEE Access*, 9, 116338-116355.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Santos, D. F. S., De Souza, G. B., & Marana, A. N. (2017). A 2D Deep Boltzmann Machine for robust and fast vehicle classification. 2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI).
- Satar, B., & Dirik, A. E. (2018). Deep Learning Based Vehicle Make-Model Classification. International Conference on Artificial Neural Networks.
- Sathyanarayana, N., & Narasimhamurthy, A. M. (2022). Vehicle Type Classification Using Hybrid Features and a Deep Neural Network. *International Journal of Applied Metaheuristic Computing (IJAMC)*, 13(1), 1-22.
- Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE International Conference on Computer Vision.
- Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. International Conference on Learning Representations.

- Shokravi, H., & Bakhary, N. (2017). Comparative analysis of different weight matrices in subspace system identification for structural health monitoring. *IOP Conference Series: Materials Science and Engineering*.
- Shokravi, H., Shokravi, H., Bakhary, N., Heidarrezaei, M., Rahimian Kolor, S. S., & Petru, M. (2020a). Application of the subspace-based methods in health monitoring of civil structures: A systematic review and meta-analysis. *Applied Sciences*, *10*(10), 3607.
- Shokravi, H., Shokravi, H., Bakhary, N., Heidarrezaei, M., Rahimian Kolor, S. S., & Petru, M. (2020b). A review on vehicle classification and potential use of smart vehicle-assisted techniques. *Sensors*, *20*(11), 3274.
- Shokravi, H., Shokravi, H., Bakhary, N., Heidarrezaei, M., Rahimian Kolor, S. S., & Petru, M. (2020c). Vehicle-assisted techniques for health monitoring of bridges. *Sensors*, *20*(12), 3460.
- Shokravi, H., Shokravi, H., Bakhary, N., Rahimian Kolor, S. S., & Petru, M. (2020d). A comparative study of the data-driven stochastic subspace methods for health monitoring of structures: A bridge case study. *Applied Sciences*, *10*(9), 3132.
- Shokravi, H., Shokravi, H., Bakhary, N., Rahimian Kolor, S. S., & Petru, M. (2020e). Health monitoring of civil infrastructures by subspace system identification method: an overview. *Applied Sciences*, *10*(8), 2786.
- Siddiqui, A. J., Mammeri, A., & Boukerche, A. (2016). Real-time vehicle make and model recognition based on a bag of SURF features. *IEEE Transactions on Intelligent Transportation Systems*, *17*(11), 3205-3219.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*.
- Sochor, J., Herout, A., & Havel, J. (2016). Boxcars: 3d boxes as cnn input for improved fine-grained vehicle recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Soon, F. C., Khaw, H. Y., Chuah, J. H., & Kanesan, J. (2018). PCANet-based convolutional neural network architecture for a vehicle model recognition system. *IEEE Transactions on Intelligent Transportation Systems*, *20*(2), 749-759.
- Soon, F. C., Khaw, H. Y., Chuah, J. H., & Kanesan, J. (2020). Semisupervised PCA Convolutional Network for Vehicle Type Classification. *IEEE Transactions on Vehicular Technology*, *69*(8), 8267-8277.

- Sotheany, N., & Nuthong, C. (2017). Vehicle classification using neural network. 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON).
- Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *International Conference on Learning Representations*.
- Sun, C. (2000). An investigation in the use of inductive loop signatures for vehicle classification. *Research Reports, Institute of Transportation Studies, PATH, University of California, Berkeley*.
- Sun, C., & Ritchie, S. G. (2000). Heuristic vehicle classification using inductive signatures on freeways. *Transportation Research Record, 1717*(1), 130-136.
- Sun, G., Cholakkal, H., Khan, S., Khan, F., & Shao, L. (2020). Fine-grained recognition: Accounting for subtle differences between similar classes. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Sun, W., Zhang, X., Shi, S., He, J., & Jin, Y. (2017). Vehicle type recognition combining global and local features via two-stage classification. *Mathematical Problems in Engineering, 2017*.
- Sun, Z., & Ban, X. J. (2013). Vehicle classification using GPS data. *Transportation Research Part C: Emerging Technologies, 37*, 102-117.
- Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
- Tafazzoli, F., Frigui, H., & Nishiyama, K. (2017). A large and diverse dataset for improved vehicle make and model recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*.
- Tajar, A. T., Ramazani, A., & Mansoorizadeh, M. (2021). A lightweight Tiny-YOLOv3 vehicle detection approach. *Journal of Real-Time Image Processing, 1-13*.

- Tamam, M., Dwiono, W., & Safian, R. (2020). Design a prototype of the application system of classification and calculating motor vehicles on highway. *IOP Conference Series: Materials Science and Engineering*.
- Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *International Conference on Machine Learning*.
- Tan, M., & Le, Q. (2021). Efficientnetv2: Smaller models and faster training. *International Conference on Machine Learning*.
- Tang, Y., Zhang, C., Gu, R., Li, P., & Yang, B. (2017). Vehicle detection and recognition for intelligent traffic surveillance system. *Multimedia Tools and Applications*, 76(4), 5817-5832.
- Tanveer, M. S., Khan, M. U. K., & Kyung, C.-M. (2021). Fine-tuning darts for image classification. *2020 25th International Conference on Pattern Recognition (ICPR)*.
- Tian, B., Yao, Q., Gu, Y., Wang, K., & Li, Y. (2011). Video processing techniques for traffic flow monitoring: A survey. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*.
- Tian, T., Li, L., Chen, W., & Zhou, H. (2021). SEMSDNet: A multiscale dense network with attention for remote sensing scene classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 5501-5514.
- Tok, A., & Ritchie, S. G. (2010). Vector Classification of Commercial Vehicles Using a High Fidelity Inductive Loop Detection System. *Proceedings of the 89th Annual Meeting Transportation Research Board, Washington, DC, USA*.
- Touvron, H., Cord, M., El-Nouby, A., Verbeek, J., & Jégou, H. (2022). Three things everyone should know about Vision Transformers. *European Conference on Computer Vision*.
- Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., & Jégou, H. (2021). Going deeper with image transformers. *Proceedings of the IEEE/CVF International Conference on Computer Vision*.
- Tripathi, J., Chaudharya, K., Joshia, A., & Jawaleb, J. (2015). Automatic vehicle counting and classification. *International Journal of Innovative and Emerging Research in Engineering*, 2(4).
- van der Maaten, L. J. P., & Hinton, G. E. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605), 5.

- Vandergriendt, C., & Zimlich, R. (2022). *An Easy Guide to Neuron Anatomy with Diagrams*. <https://www.healthline.com/health/neurons>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Neural Information Processing Systems*.
- Wang, C.-Y., Yeh, I.-H., & Liao, H.-Y. M. (2021). You only learn one representation: Unified network for multiple tasks. *Journal of Information Science and Engineering*, 39, 691-709.
- Wang, G., Cheng, L., Lin, J., Dai, Y., & Zhang, T. (2021). Fine-grained classification based on multi-scale pyramid convolution networks. *PloS one*, 16(7), e0254054.
- Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., & Gong, Y. (2010). Locality-constrained linear coding for image classification. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- Wang, L., He, K., Feng, X., & Ma, X. (2022). Multilayer feature fusion with parallel convolutional block for fine-grained image classification. *Applied Intelligence*, 52(3), 2872-2883.
- Wang, P., Cao, Y., & Lu, L. (2022). A Novel Part Feature Integration and Fusion Method for Fine-Grained Vehicle Recognition. ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).
- Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020). ECA-Net: efficient channel attention for deep convolutional neural networks, 2020 IEEE. CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.
- Wang, X., Ma, X., & Grimson, W. E. L. (2008). Unsupervised activity perception in crowded and complicated scenes using hierarchical bayesian models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3), 539-555.
- Wang, X., Zhang, W., Wu, X., Xiao, L., Qian, Y., & Fang, Z. (2019). Real-time vehicle type classification with deep convolutional neural networks. *Journal of Real-Time Image Processing*, 16(1), 5-14.
- Wang, Y., Ban, X., Wang, H., Wu, D., Wang, H., Yang, S., . . . Lai, J. (2019). Detection and classification of moving vehicle from video using multiple spatio-temporal features. *IEEE Access*, 7, 80287-80299.
- Wei, H., Liu, H., Ai, Q., Li, Z., Xiong, H., & Coifman, B. (2013). Empirical innovation of computational dual - loop models for identifying vehicle classifications against

varied traffic conditions. *Computer - Aided Civil and Infrastructure Engineering*, 28(8), 621-634.

Wei, X.-S., Song, Y.-Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., . . . Belongie, S. (2021). Fine-grained image analysis with deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wen, X., Shao, L., Xue, Y., & Fang, W. (2015). A rapid learning algorithm for vehicle classification. *Information Sciences*, 295, 395-406.

Wen, Y., Zhang, K., Li, Z., & Qiao, Y. (2016). A discriminative feature learning approach for deep face recognition. *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14*.

Wieczorkowska, A., Kubera, E., Słowik, T., & Skrzypiec, K. (2018). Spectral features for audio based vehicle and engine classification. *Journal of Intelligent Information Systems*, 50(2), 265-290.

Won, M. (2020). Intelligent traffic monitoring systems for vehicle classification: A survey. *IEEE Access*, 8, 73340-73358.

Won, M., Sahu, S., & Park, K.-J. (2019). Deepwittraffic: Low cost wifi-based traffic monitoring system using deep learning. 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS).

Woo, S., Park, J., Lee, J.-Y., & So Kweon, I. (2018). Cbam: Convolutional block attention module. *Proceedings of the European Conference on Computer Vision (ECCV)*.

Wu, H., Guo, H., Miao, Q., Huang, M., & Wang, J. (2022). Graph Neural Networks Based Multi-granularity Feature Representation Learning for Fine-Grained Visual Categorization. *International Conference on Multimedia Modeling*.

Wu, L., & Coifman, B. (2014). Vehicle length measurement and length-based vehicle classification in congested freeway traffic. *Transportation Research Record*, 2443(1), 1-11.

Xiang, Y., Fu, Y., & Huang, H. (2019). Global topology constraint network for fine-grained vehicle recognition. *IEEE Transactions on Intelligent Transportation Systems*, 21(7), 2918-2929.

Xie, J., Zheng, Y., Du, R., Xiong, W., Cao, Y., Ma, Z., . . . Guo, J. (2021). Deep Learning-Based Computer Vision for Surveillance in ITS: Evaluation of State-of-the-Art Methods. *IEEE Transactions on Vehicular Technology*, 70(4), 3027-3042.

- Xie, X., Zhou, Y., & Kung, S.-Y. (2020). Exploring Highly Efficient Compact Neural Networks For Image Classification. 2020 IEEE International Conference on Image Processing (ICIP).
- Xu, Y., Zhang, Z., Zhang, M., Sheng, K., Li, K., Dong, W., . . . Sun, X. (2021). Evo-vit: Slow-fast token evolution for dynamic vision transformer. AAAI Conference on Artificial Intelligence.
- Yang, L., Luo, P., Change Loy, C., & Tang, X. (2015). A large-scale car dataset for fine-grained categorization and verification. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.
- Yang, S., Zhang, J., Bo, C., Wang, M., & Chen, L. (2019). Fast vehicle logo detection in complex scenes. *Optics & Laser Technology*, 110, 196-201.
- Yang, W.-D., Gao, Z.-M., Wang, K., & Liu, H.-Y. (2016). A privacy-preserving data aggregation mechanism for VANETs. *Journal of High Speed Networks*, 22(3), 223-230.
- Yao, Y., Tian, B., & Wang, F.-Y. (2016). Coupled multivehicle detection and classification with prior objectness measure. *IEEE Transactions on Vehicular Technology*, 66(3), 1975-1984.
- Yousaf, K., Iftikhar, A., & Javed, A. (2012). Comparative analysis of automatic vehicle classification techniques: a survey. *International Journal of Image, Graphics and Signal Processing*, 4(9), 52.
- Yu, Y., Jin, Q., & Chen, C. W. (2018). FF-CMnet: A CNN-based model for fine-grained classification of car models based on feature fusion. 2018 IEEE International Conference on Multimedia and Expo (ICME).
- Yu, Y., Liu, H., Fu, Y., Jia, W., Yu, J., & Yan, Z. (2022). Embedding Pose Information for Multiview Vehicle Model Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Yu, Y., Xu, L., Jia, W., Zhu, W., Fu, Y., & Lu, Q. (2020). CAM: A fine-grained vehicle model recognition method based on visual attention model. *Image and Vision Computing*, 104, 104027.
- Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., & Wu, W. (2021). Incorporating convolution designs into visual transformers. Proceedings of the IEEE/CVF International Conference on Computer Vision.

- Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. British Machine Vision Conference.
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. European Conference on Computer Vision.
- Zhai, X., Kolesnikov, A., Houlsby, N., & Beyer, L. (2022). Scaling vision transformers. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.
- Zhang, B. (2012). Reliable classification of vehicle types based on cascade classifier ensembles. *IEEE Transactions on Intelligent Transportation Systems*, 14(1), 322-332.
- Zhang, F., Li, M., Zhai, G., & Liu, Y. (2021). Multi-branch and multi-scale attention learning for fine-grained visual categorization. International Conference on Multimedia Modeling.
- Zhang, J., Bargal, S. A., Lin, Z., Brandt, J., Shen, X., & Sclaroff, S. (2018). Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126(10), 1084-1102.
- Zhang, L., Huang, S., & Liu, W. (2021). Enhancing mixture-of-experts by leveraging attention for fine-grained recognition. *IEEE Transactions on Multimedia*, 24, 4409-4421.
- Zhang, Q., Zhuo, L., Zhang, S., Li, J., Zhang, H., & Li, X. (2018). Fine-grained vehicle recognition using lightweight convolutional neural network with combined learning strategy. 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM).
- Zhang, T., Chang, D., Ma, Z., & Guo, J. (2021). Progressive co-attention network for fine-grained visual classification. 2021 International Conference on Visual Communications and Image Processing (VCIP).
- Zhao, J., Hao, S., Dai, C., Zhang, H., Zhao, L., Ji, Z., & Ganchev, I. (2022). Improved Vision-Based Vehicle Detection and Classification by Optimized YOLOv4. *IEEE Access*.
- Zhao, P., Li, Y., Tang, B., Liu, H., & Yao, S. (2023). Feature relocation network for fine-grained image classification. *Neural Networks*, 161, 306-317.

- Zheng, H., Fu, J., Mei, T., & Luo, J. (2017). Learning multi-attention convolutional neural network for fine-grained image recognition. *Proceedings of the IEEE International Conference on Computer Vision*.
- Zhu, M., Wan, S., Jin, P., & Tian, Q. (2021). A Feature Fusion Method Based on Multi-Classification Losses for Fine-Grained Visual Categorization. *2021 IEEE International Conference on Big Data (Big Data)*.
- Zhu, Q., & Li, Z. (2022). Data Augmented Dual-Attention Interactive Image Classification Network. *International Conference on Artificial Neural Networks*.
- Zhu, Z., & Guo, Y. (2012). Vehicle style recognition based on image processing and neural network. In *Advances in Computer Science and Information Engineering* (pp. 1-8). Springer.
- Zhuang, P., Wang, Y., & Qiao, Y. (2020). Learning attentive pairwise interaction for fine-grained classification. *Proceedings of the AAAI Conference on Artificial Intelligence*.