ADAPTIVE GLOBAL REASONING WITH MULTIPLE KNOWLEDGE GRAPHS FOR OBJECT DETECTION

TAO BO

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

ADAPTIVE GLOBAL REASONING WITH MULTIPLE KNOWLEDGE GRAPHS FOR OBJECT DETECTION

TAO BO

DISSERTATION SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF COMPUTER SCIENCE

FACULTY OF COMPUTER SCIENCE & INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Tao Bo

Matric No: 17006936/1

Name of Degree: Master of Computer Science

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):

Adaptive Global Reasoning with Multiple Knowledge Graphs for Object Detection

Field of Study: Computer vision

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 University of 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:

Subscribed and solemnly declared before,

Witness's Signature Date:

Name:

Designation:

ADAPTIVE GLOBAL REASONING WITH MULTIPLE KNOWLEDGE GRAPHS FOR OBJECT DETECTION

ABSTRACT

The dominant object detection system's mechanism is to propose some regions of interest and then classify them and locate these regions with bounding boxes. In other words, the current object detection system is modeled as classification on boxes in parallel without considering the relationship between the objects. Such strong semantic information should be used to help current object detection systems to get more accurate results. In contrast, human vision recognition system can recognize objects easily, even in very complex scenes (heavy occlusion, more categories, class ambiguities, etc.). The main reason is that humans have the knowledge (common sense) to help them recognize what they see. When humans can not see the target object clearly, the visual reasoning process goes on: With the help of surrounding objects and environment or context, humans usually have the ability to deduce the object. Inspired by the human visual recognition mechanism, many works have been done to incorporate knowledge base to current object detection system to imitate the reasoning process. The dominant reasoning process is to propagate region features through a fixed external knowledge graph. The nodes in the graph represent region proposals, and edges represent connections or relationships of each pair of nodes. After the learning process through the knowledge graph, the region proposals' features will be enhanced, and the accuracy of the final prediction on these region proposals will be higher. The current state-of-the-art object detection with reasoning process called Reasoning RCNN (H. Xu et al., 2019) is to propagate the global semantic classes feature in a single knowledge graph. However, one handcraft knowledge graph that only considers a single factor is not general enough. It cannot fit each image very well because there is a semantic gap between an individual image and external linguistic contexts. Assume two objects do not show up in an image

simultaneously, while the external knowledge graph has strong a relationship between those two objects, it will be difficult to classify these two objects correctly in this image. On account of this problem, the global semantic classes feature will be propagated through multiple knowledge graphs to get a more general and robust feature representation. In this research, the attribute graph and co-occurrence graph with learning parameters will be used to make the relationships between each pair of classes more general and robust. By adding the reasoning module without changing the whole neural network architecture, the proposed method is lightweight and flexible.

Keywords: Object detection, Reasoning, Knowledge Graphs.

PENALARAN GLOBAL ADAPTIF DENGAN GRAF PENGETAHUAN PELBAGAI UNTUK PENGESANAN OBJEK

ABSTRAK

Objektif mekanisme sistem pengesanan objek adalah langkah pertama untuk mengusulkan beberapa zon yang menarik dalam imej. Kemudian, sistem akan mengklasifikasikannya dan melukis kotak pengikat di sekelilingnya. Oleh itu, sistem pengesanan objek semasa dimodelkan sebagai klasifikasi zon secara selari tanpa mempertimbangkan hubungan antara objek. Maklumat semantik ini harus digunakan untuk membantu sistem pengesanan objek untuk memberi prestasi lebih maju. Justeru, sistem penglihatan manusia dapat mengenali objek dengan mudah walaupun dalam pemandangan atau imej yang rumit seperti halangan lintang penglihatan, kategori objek berlebihan, kekaburan kategori objek dan lain-lain. Hal ini demikian kerana sistem penglihatan manusia mempunyai kemampuan untuk mengenali hubungan semantik diantara objek secara tidak langsung yang menolong pengenalan objek. Manusia biasanya mengenal objek dengan konteks alam sekitarnya. Diilhamkan oleh mekanisme penglihatan manusia, banyak penambahan telah dilakukan untuk membaikkan sistem pengesanan objek. Proses penaakulan yang efektif adalah sistem yang menyebarkan ciri-ciri zon melalui graf pengetahuan luaran yang tetap. Nod dalam graf mewakili cadangan zon penaakulan, dan pinggir-pinggir graf mewakili hubungan atau hubungan antara setiap pasangan kategori. Selepas proses pembelajaran melalui graf pengetahuan, prestasi ciri-ciri cadangan zon imej akan ditingkatkan dan penaakulan zon-zon ini akan mempunyai ketepatan yang lebih tinggi. Pengesanan objek terkini dengan proses penaakulan *Reasoning RCNN* (H. Xu et al., 2019) iaitu menyebarkan ciri-ciri golongan semantik global dalam graf pengetahuan tunggal. Walau bagaimanapun, satu graf pengetahuan yang hanya mempertimbangkan satu faktor tidak cukup umum. Ia tidak sesuai untuk setiap imej oleh kerana jurang semantik di antara konteks visual dan linguistik. Sekiranya

dua objek tidak muncul dalam imej secara serentak, walaupun graf pengetahuan luaran yang mempunyai hubungan yang kuat antara ke dua-dua objek tersebut, sistem tidak boleh mengklasifikasikan objek tersebut dengan tepat. Malah, ciri-ciri kelas semantik global akan disebarkan melalui beberapa graf pengetahuan untuk mendapatkan perwakilan ciri-ciri yang lebih umum dan mantap. Dalam penyelidikan ini, graf atribut dan graf Co-Occurence dengan parameter pembelajaran akan digunakan untuk menjadikan hubungan antara setiap pasangan kategori menjadi lebih umum dan mantap. Dengan menambahkan modul penaakulan dengan tanpa mengubah keseluruhan reka bentuk rangkaian neural, kaedah kami adalah ringan dan fleksibel.

Kata kunci: Pengesanan objek, Berakal, Graf Pengetahuan.

ACKNOWLEDGEMENTS

The two-year master's career is coming to an end. Looking back on the past two years, I am full of emotion and gratitude. First, I sincerely appreciate my supervisor, Associate Prof. Chan Chee Seng, who took me into the computer vision and deep learning area and guided me through my whole academic career at the University of Malaya. Thank him for his academic support at every stage of my research, for taking the time to discuss how to conduct my research, and for his every email reply, no matter how busy he is. I also want to give my thanks to Dr. Hoo Wai Lam, who gives me advice on conducting experiments with limited hardware resources, correcting a lot of the errors in my thesis. Without their enthusiasm, knowledge, and patience, I couldn't finish this research project.

Secondly, I wish to express sincere gratitude to my family for their particular concern and help. Thank them for their spiritual and financial support. Whenever I encounter difficulties in my academic research and real life, they are always on my side, and their encouragement is my motivation.

In the last, I would like to thank the University of Malaya for providing me such an excellent chance to expand my horizons, enrich my knowledge and learn about different cultures. During my two-year graduate study career, the excellently designed lectures gave me more new knowledge. I also want to thank those officers who helped me, and I could not move on so far without their help. I am very proud to be a student here.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

AP	:	Average Precision
CNN	:	Convolution Neural Network
FC	:	Fully Connected Layer
HOG	:	Histogram of Oriented Gradients
IoU	:	Intersection over Union
JS	:	Jensen-Shannon
LSVRC	:	ImageNet Large Scale Visual Recognition Challenge
mAP	:	Mean Average Precision
NMS	:	Non-Maximum Suppression
R-CNN	:	Region proposal with Convolution Neural Network
ReLU	:	Rectified Linear Unit
ResNet	:	Residual Networks
ROI	:	Region of Interest
RPN	:	Region Proposal Network
SENet	:	Squeeze-and Excitation network
SPPNet	:	Spatial Pyramid Pooling Network
SSD	:	Single Shot Multibox Detector
VG	:	Visual Genome
YOLO	:	You Only Look Once
PASCAL VOC	:	PASCAL Visual Object Classes

CHAPTER 1: INTRODUCTION

1.1 Generic object detection

Object detection, which is one of the most important and challenging computer vision tasks, aims to detect visual objects of a specific class and location in images or videos. The boxes usually represent the location of the object. The classic strategy for object detection is to classify each box, so the object detection problem is also modeled as a machine learning problem. In each image, the classification rule is to map each box to several foreground object classes or background classes. The classifier is trained to learn this mapping. Although this classic strategy brings a lot of conveniences, it also comes with many problems. Among all, one of the most problems is there are too many boxes in the image needed to be classified.

1.1.1 Challenges and problems in object detection

The output space of the object detection is infinite, thus, instead of working with infinitely possible boxes, finite boxes are quantized to represent the original continuous space. The proxy problem will involve classifying each of the quantized boxes rather than the original boxes. Naturally, the quantized boxes do not correspond exactly to the original boxes. Therefore, the location of the detection is inaccurate. The distance between the original boxes and quantized boxes is the quantization error introduced by the approximation. To address the loss of localization accuracy, an additional task regressor is added to the model. The model will be trained to predict the quantization error. To solve the problem of too many boxes, the proxy problem is switched to the classification and regression tasks.

Because of the proxy problem, the classification rule of object detection is not well defined. Instead of outputting the exact label of the ground truth box, a set of quantized boxes are classified. Therefore, the label assignment rule needs to be specified, and the foreground label will be assigned to a group of quantized boxes. The standard solution is to find label heuristics such as Intersection over Union (IoU), centeredness and containment, etc. Underlying this classification rule, there are several consequences. For example, there may be no more than one quantized box is correct; Independent predictions for each quantized box; Require a set of operations to resolve redundancy such as Non-Maximum Suppress algorithm (R. Rothe et al., 2014) (NMS).

The last problem is the imbalance between foreground and background boxes. The foreground objects in an image are finite, but background objects are infinite. As it is known to all that training the classification model on imbalanced data can be difficult. The classifier can simply ignore the minority class, resulting in bad performance of object detection. Furthermore, the processing speed will slow down because the classifier will spend more time in the computation on the background boxes. To solve this problem, the loss function was modified, such as focal loss (T. Lin et al., 2017), to pay more attention to the hard training example. The fine-turning classifier Cascade RCNN (Z. Cai & N. Vasconcelos, 2019) retains true positives and rejects false positives to generally alleviates imbalance data.

1.1.2 Milestones in object detection

Based on the classic framework discussed above, there have been many brilliant works trying to solve the problem and improve the system performance in both accuracy and speed. Moreover, due to its wide range of applications such as face detection, target tracking, automatic drive, and recent breakthroughs in deep learning, it has attracted more and more researchers to set foot in this area. In the last two decades, generic object

detection has gained significant improvement, and it can be divided into two periods: the traditional-method period and the deep-learning period.

For most of the early traditional-method-based works, their algorithms were built based on the handcrafted features due to the lack of efficient feature representation. In order to get better accuracy and speed, they had to design sophisticated and complicated feature representation and varieties of speed-up skills. A lot of algorithms such as Viola Jones Detectors (P. Viola & M. Jones, 2001), Histogram of Oriented Gradients (HOG) (N. Dalal & B. Triggs, 2005), Deformable Part-based Model (P. Felzenszwalb et al., 2008) were designed in this period which all achieved enormous success. However, as the limitation of handcrafted feature, the performance of object detection has reached a bottleneck. Thanks to the significant improvement of the image classification by deep convolution neural network (CNN) (A. Krizhevsky et al., 2012), the icebreaker R-CNN (R. Girshick et al., 2014) came out and lead the object detection evolution in leaps and bounds.

According to different region proposal methods, most of the existing CNN-based object detection systems could be divided into two categories: one-stage strategy and two-stages strategy. For those one-stage strategy methods, the most commonly used and remarkable detection systems such as You Only Look Once (YOLO) (J. Redmon et al., 2016), Single Shot Multibox Detector (SSD) (W. Liu et al., 2016), and RetinaNet (T.-Y. Lin et al., 2018). For those two-stage strategy detectors, the mainstream is R-CNN (R. Girshick et al., 2014) series: Fast Region-based Convolutional Network (Fast R-CNN) (R. Girshick, 2015), Faster R-CNN (S. Ren et al.,2015), and Feature Pyramid Network (FPN) (T.-Y. Lin et al., 2017). The main idea of these two strategy methods is to propose several regions of interest (ROI) and then perform classification and bounding box regression on these ROIs.

While in a one-stage strategy system, they don't have a region proposal step, they propose predicted boxes from input images directly and classify these bounding boxes. Comparing the speed and accuracy of these systems' performance, generally, those two-stage strategy methods could be more accurate, and the speed of one-stage strategy methods is higher.

1.2 Visual inference with prior knowledge

Although the performance of two kinds of object detection algorithms has been improved by the better network architecture design (from RCNN to Faster RCNN) and fully annotated datasets such as PASCAL VOC (M. Everingham et al.,2010), MS COCO (O. Vinyals et al., 2016), there are still a lot of challenges that needed to be solved in the complex scenarios such as large-scale object detection (a lot of different categories exist in one image simultaneously). The problems caused by class ambiguities, heavy occlusion, and tiny size objects can occur frequently. The scarcity of the data in some classes (rare categories) makes it harder for current networks to learn and detect effectively, the main reason is that current the-state-of-art object detection algorithms do visual recognition in region proposals separately. To solve these problems, high-quality feature representation for each object along with sufficient and balanced foreground-background data are required to train the network. However, the datasets for object detection are either in small scale or part of bounding boxes are annotated such as ImageNet (O. Russakovsky et al., 2015), and it is very tedious and impractical to label each category in a large-scale context. The human annotation cost and the imbalanced categories dataset significantly limit the performance of current object detection systems.

In contrast, the human visual system can recognize objects easily without learning many examples. Humans can easily understand huge numbers of concepts in our world even if these objects are ambiguous (apple can be green or red), have heavy occlusion or overlap, rare categories, and tiny size objects (small boats on the sea). One of the reasons is that humans have prior knowledge and reasoning ability. Humans recognize objects not only rely on the visual appearance but also based on the knowledge of the world (common sense), which human learns from experience and language. For example, when people see a tiny monitor, they do the recognition based on their appearance (rectangle, metal material, etc.) and recall their knowledge to search similar appearance categories. Then people do reasoning by combining the knowledge with semantic coherency: this small rectangle metal object was installed on top of the bank gate for security monitoring, so it can be concluded that it is a monitor. To sum up, making the machine more intelligent and enable it to incorporate prior knowledge in computer vision is a critical task.

1.3 Incorporate knowledge in the object detection system

Recently, the end-to-end training style and fully annotated datasets have shown great significance in many natural language processing and computer vision tasks. However, when models require reasoning, incorporating prior knowledge in the network along with the end-to-end training process can introduce better inductive bias beyond what is provided in the training dataset. When LeCun et al. (1989) brought the convolution layer into the neural network, he also implied that it could not achieve good generalization performance in the real-world application unless some prior knowledge on a specific task is embedded into the system. But what is the prior knowledge for machine learning or deep learning models?

Based on the structure of the current neural network, there are several ways to encode prior knowledge. (1) Concatenate the hand-craft features to the original feature representations of the deep neural networks. It is also known as feature enhancement, and it could be done in different places or periods of feedforward propagation. (2) Use probability distribution of prior knowledge to reweight softmax output of the classifier. In the deep neural network, prior knowledge could be considered as a probability distribution before the model learning from the dataset. (3) Reweight loss function according to the importance of different classes. It could be obtained from prior knowledge so that the model can better recognize the difficult classes. Although prior knowledge seems random and difficult to organize, there has been a lot of research (X. Chen et al., 2013; F. Sadeghi et al., 2015) incorporated knowledge graphs into visual recognition.

In the visual reasoning domain, a knowledge graph is mostly used to structure the prior knowledge. A knowledge graph is a graph that models semantic consistency between each entity, where each node represents an entity, and each edge represents relationships between two entities. A knowledge graph can be built automatically or manually by the linguistic information distilled from the annotated dataset. In order to make the current mainstream object detection system more intelligent and can imitate human's reasoning procedure, most of the early works (Yong Liu et al., 2018; K. Marino et al., 2017; Y. Seo et al., 2017; Y. Li et al., 2016; D. Teney et al., 2017; H. Hu et al., 2016) treat object detection as a problem of graph structure inference where the objects are treated as nodes and relationship between objects are modeled as edges in such a graph. More specifically, they iteratively propagate the information from the visual features of nodes, scene context, and spatial and visual relationships between nodes to adjust knowledge graphs, which are used to predict object category and bounding box offsets. Those knowledge graphs are built based on the information distilled from the

image convolution feature maps, which is not external knowledge like human common sense. To solve this problem, the iterative visual reasoning approach (X. Chen et al., 2018) adds a fixed prior knowledge for global reasoning, which propagates the relationships of each pair of region nodes. However, all the methods above only consider region proposal features to propagate throughout the knowledge graphs, which the reasoning results could still fail when the bad image feature representation happened such as heavy occlusion, class ambiguities especially in the large-scale object detection scenario. To avoid directly propagating the region proposal visual features, a recent work called Reasoning RCNN (H. Xu et al., 2019) choose to build a global semantic pool to represent the features of all categories. They use an external prior knowledge graph which is built from the linguistic annotation of Visual Genome (VG) (R. Krishna et al., 2017) dataset to propagate linguistic information among all the categories in the global semantic pool. Last, they use a soft-mapping mechanism to make a global semantic pool link to region proposal nodes. With related categories' information aggregating, the reasoning process in the object detection system was improved to a certain extent.

1.4 Objectives and statement of the problem

The current global knowledge graph of Reasoning RCNN is built from the annotation of the VG dataset. The relationship between each pair of classes is based on a single factor: co-occurrence frequency or attribute distribution. However, only considering one factor to build a knowledge base is not general and robust enough, when compared with the complicated knowledge base of humans. Moreover, the construction methods of the knowledge graph also have shortcomings. For example, they use Jensen-Shannon (JS) divergence to measure similarities between two categories' attribute distributions. The similarities value range is from zero to one, which will make the variance of the

distribution too small. In the end, the final predictions on region nodes will be affected by some unrelated region nodes' information. For the co-occurrence frequency knowledge graph, they count the frequency of co-occurrence of each pair of categories, the higher the frequency, the higher the similarity. At the same time, overall statistics cannot fit each image context very well. It only works for those pairs of counted categories but does not generalize to the unknown categories in the new image, which means if two categories do not exist in the VG dataset, the numerical semantic consistency between them will be zero. Thus, it will not be helpful.

With these problems, it can be found out that one fixed handcraft knowledge graph which only considering a single factor can't fit each image very well due to the semantic gap between linguistic and individual visual context, and the global knowledge graph is fully connected, which it will be affected by redundant and noisy information from irrelevant objects.

As noticing the problem in the current Reasoning RCNN, what needs to be done is to reduce the semantic gap and make knowledge graph reasoning more accurate and robust. There are two ways to do that. One way is to build a more complex and robust knowledge graph that includes more factors by a sophisticated construction method. Another way is to incorporate more single factor knowledge graphs, with learning parameters controlling each knowledge graph's weight. At the end of that, the final relationship between each pair of classes will be adjusted, and the knowledge base will be more general and robust. In this research, instead of finding a sophisticated way to construct an elaborate knowledge graph, the second way is used to extend a single knowledge graph to multiple knowledge graphs. To evaluate if the reasoning module with multiple knowledge graphs is better than the current reasoning module with a single knowledge graph, the mean average precision of the two models will be compared.

1.5 The significance of the study

The human visual recognition system is so powerful that it is beyond any current deep-learning-based object detection system. One of the reasons is that human has a strong and complex knowledge base. With the help of the reasoning process, people can easily recognize the objects even in a complex scenario. In order to achieve better performance and cultivate the reasoning ability of object detection systems, it is essential to build an efficient, robust, and general knowledge base. It is not only crucial for object detection but also other visual recognition problems such as image classification, instance segmentation, and target tracking, etc. Furthermore, through this study, the role of knowledge graph playing in the reasoning process at the deep neural network can be better understood, so that other methods of how to build a knowledge base for object detection systems can be explored and further improvement of object detection can be achieved from different perspectives.

CHAPTER 2: LITERATURE REVIEW

This research method is based on deep learning, and the backbone CNN architecture will be briefly reviewed in chapter 2.1. After that, several famous and highly influential object detection systems based on CNN will be reviewed and compared in chapter 2.2. In chapter 2.3, the question of how to incorporate reasoning process with knowledge graph in object detection system will also be analyzed.

2.1 Convolutional neural network

As the most representative and iconic deep neural network, CNN has shown great success in computer vision and image processing tasks. CNN's basic structure consists of one input layer, several convolutional layers, pooling layers or downsampling layers, fully connected layers, and one output layer.

The role of the convolutional layer is to extract the features of the image. The convolution kernel (filter) is like a sliding window, which slides back and forth in the entire input image with a specific step size. After convolution operation, the input image's feature map will be obtained. The feature map consists of local features extracted by the convolution layer, and this convolution kernel shares parameters. The convolution kernel weights will not stop keeping updated in the training process until the training process is completed. Weight sharing of convolution kernel ensures that each pixel has a weight coefficient, but the entire picture shares these coefficients, which vastly reduces parameter numbers and the network's complexity. The convolution operation can make use of the local correlation of image space to extract features automatically because each convolution kernel can only distill one type of feature. In order to increase the

convolutional neural network's expressive ability, multiple convolution kernels need to be set.

Downsampling is another crucial concept of convolutional neural networks, which is also commonly referred to as pooling. The most common methods are maximum pooling, minimum pooling, and average pooling. With downsampling or pooling operations, the image's resolution will be reduced, and the entire network will not be easy to overfit. A convolutional layer plus a pooling layer is called a feature extraction unit. These feature extraction units will appear multiple times in the deep CNN. However, not every convolutional layer is followed by a pooling layer. Most of CNN only has three pooling layers. At the end of the network, there are generally one or two fully connected layers that are responsible for connecting the extracted feature maps. Finally, the final classification result is obtained through the classifier.

The first CNN was designed by LeCun et al. (1989), who combined a backpropagation algorithm with weights sharing convolutional layers to create the convolutional neural network. It successfully applied the convolutional neural network to the handwritten character recognition system. After several years, they continued to improve their model and proposed LeNet-5 (LeCun et al., 1998), which is the first classic architecture of convolutional neural networks. It significantly increased the accuracy of handwritten character recognition.

Krizhevsky et al. (2012) used the expanded depth of CNN called AlexNet to achieve the best classification accuracy in the ImageNet Large Scale Visual Recognition Challenge (LSVRC). Except for increasing the depth of the network, AlexNet also used many new technologies. Rectified linear unit (ReLU) was used to replace the saturated nonlinear function TANH function, which reduces the computational complexity. The training speed of the model has also been increased several times. The Dropout technique

was used in the training period, which makes some random neurons in the middle layer set to 0. It makes the model more robust, at the same time it reduces the overfitting of the fully connected layers. They also increased the number of training examples through image translation, horizontal flipping, and changing image grayscale to reduce overfitting.

(Simonyan et al., 2014) proposed a model called VGG network and discussed the influence of "depth" for deep CNN architecture. Based on the AlexNet, they replaced large receptive fields like 7*7 with smaller receptive fields 3*3 to make decision functions more discriminative. In order to find out the influence of the depth of CNN, they continued adding convolutional layers with 3*3 convolution kernels on the top of each layer. Their experimental results showed that when the number of layers in the network reaches from 16 to 19, the model's performance on accuracy can be effectively improved. However, their model is tough and slow to train.

The depth of architecture of CNN is going to larger and larger since VGG network. However, simply stacking the layers in networks does not increase the performance. Furthermore, it is very hard to train a deep neural network because of the vanishing gradient problem. (He et al., 2016) used Residual Networks (ResNet) to solve gradient disappearance when the depth of CNN is too large. The main feature of ResNet is the cross-layer connection. It introduces shortcut connections to add the input across layers with convolution results. With the residual union, hundreds or thousands of layers in ResNet could be fully trained.

In recent years, many researchers were attracted by CNN's outstanding characteristics such as local connection, weight sharing, pooling operation, etc. With the characteristic of weight sharing, CNN can significantly reduce the numbers of weights that need to be trained and the network's computational complexity. At the same time, the pooling

operation makes the network have a certain kind of translation invariance and scaling invariance to the local transformation of the input image, and it dramatically improves the generalization ability of the network. It directly inputs original images into the network and then implicitly learns from the training data. Thus, it can avoid many drawbacks of manually extracting features which could lead to error accumulation. Its entire classification process is automatic. Although these good characteristics have made CNN widely used in various fields, there are still many problems in CNN that need to be solved. The downsampling or pooling layer could result in a lot of valuable information loss. The backpropagation algorithm makes it hard to train a deep neural network effectively. Although image classification tasks can perform very well due to the excellent feature extraction ability of deep convolutional neural networks, some problems such as occlusion or overlap still cannot work out well.

2.2 CNN based object detection

When reviewing the development of the CNN based object detection system in recent few years, several two-stage object detectors are identified as milestones in this domain, such as Fast RCNN (Girshick, 2015), Faster RCNN (S. Ren et al., 2015), FPN (T.-Y. Lin et al., 2017) and one-stage object detectors such as YOLO (J. Redmon et al., 2016), SSD (W. Liu et al., 2016), etc. Furthermore, with the foundation built by these brilliant works, the research in object detection expanded exponentially. Instead of building a new architecture from scratch, which is very expensive and hard to design and train the network, most of the new object detection systems were developed based on these early works, and further to improve the accuracy of classification and localization. By following this tradition of object detection research, the system is designed based on state-of-the-art object detectors, which could be both two-stage and one-stage. In the following

paragraphs, both mainstream two-stage and one-stage object detection systems will be reviewed.

2.2.1 Two-stage frameworks

With CNN's good performance in image classification, naturally, researchers start to consider if CNN could be used to solve the object detection problem. In 2014, RCNN (R. Girshick et al.) was proposed. They first adopt a selective search algorithm (J. R. Uijlings et al., 2013) to generate about 2k region proposals for each Image. After warping each region proposal to a fixed size, 2k fixed size region proposals are fed into CNN model AlexNet which is pre-trained in the large-scale dataset such as ImageNet to get feature representation. Finally, with these extracted features, Support Vector Machine (SVM) classifiers are used to predict an object's presence within each region and recognize object categories, and the bounding box regression produces final bounding boxes for object location. Although it obtained a significant improvement, the drawbacks are apparent: the feature computation process on each region proposal is separated, making multiple CNN modes hard to train and optimize, and detection speed is also extremely slow. Additionally, as the CNN fully connected layers require a fixed-size image input, they choose to warp each region proposal to a fixed size. However, the geometric distortion could lead to feature loss due to this warping operation, and the accuracy would be reduced.

To solve the problem mentioned above, K. He et al. (2014) proposed Spatial Pyramid Pooling Networks (SPPNet). SPPNet mainly introduces a Spatial Pyramid Pooling (SPP) layer after the final convolution layer (conv5), which partitions the images from finer to coarser scales and aggregates quantized local features into mid-level representations. Unlike the RCNN, SPPNet computes convolution feature maps only once, thus it can

avoid redundant computation. Moreover, with fixed-size feature representation from the SPP layer, it could be directly fed into the CNN model's fully connected layer without needing warping operation on region proposals. Although SPPNet improves the speed of the RCNN, there are still some drawbacks. Firstly, the training process takes the same multi-stage pipeline as RCNN, which still needs to train and optimize multiple classifiers and bounding box regressions. Secondly, SPPNet only fine-tunes its fully connected layers while simply ignores all previous layers.

Next year, R. Girshick (2015) proposed a novel CNN architecture called Fast RCNN to further improve the RCNN and SPPNet. Like the SPP layer, a region of interest (RoI) pooling layer was proposed to extract the fixed size of each region proposal's feature vector, which has only one pyramid level. Different from the SPPNet, the Fast RCNN only train and optimize one classifier and bounding box regressor. The fixed-size feature vectors are fed into a sequence of FC layers and then branched into two sibling output layers: One output layer produces softmax probability estimates over foreground object classes plus a background class, another layer outputs four real-valued numbers for each of object classes. With the multi-task loss and training in a single stage, the efficiency and the accuracy of the network are improved significantly. However, Fast RCNN is still replied on the external region proposals whose computation still limits the speed of the whole system.

To solve this problem of Fast RCNN, Ren et al. (2016) introduced Region Proposal Network (RPN) to generate region proposals. They first generate K anchors (or boxes) of different scales and aspect-ratios on each convolutional feature map. Then these anchors are fed into two fully connected layers to obtain region proposals. The convolutional feature map in RPN is also shared with the Fast RCNN network. Thus, it is highly

efficient. With the simple improvement, Faster RCNN becomes the first end-to-end training, nearly real-time object detector.

An image pyramid is widely used for scale-invariant object detection of handcrafted feature methods, but deep learning avoids pyramid representation because of the high computation and memory cost. Thus, most of the deep-learning-based detectors processed detection on the last layer of the network. Lin et al. (2017) propose a Pyramid Feature Network which leverages the inherent multi-scale pyramid hierarchy of the convolutional neural network to construct a top-down feature pyramid that has high-level semantics at all scales. Their work shows significant improvement in several detectors without sacrificing efficiency. Therefore, it is generally used for both two-stage methods and one-stage methods.

In object detection, the threshold value of Intersection over Union (IoU) is used to determine the positive proposals and negative proposals, and the positive and negative sample ratio in the training period would determine the quality of the system. In most RCNN series, a low IoU threshold value like 0.5 was used to determine if the region proposals are positive or negative in the RPN. However, the low IoU threshold usually produces close negative detections in the testing period. In order to find out how IoU thresholds would affect the object detection system and to achieve higher performance, (Z. Cai & N. Vasconcelos, 2019) proposed Cascade RCNN, which trains a sequence of detectors with increasing IoU thresholds. The output of the first detector will be the training set for the next detector. With the progress of training sets, the last detector will give the best performance.

2.2.2 One-stage frameworks

Although the two-stage object detection systems have been improved significantly in both accuracy and efficiency, the computation is still too high to run the systems on limited hardware devices such as mobile phones. Thus, the first lightweight and highly efficient object detector called YOLO (J. Redmon et al., 2016) appeared. Unlike region-based object detectors that firstly generate region proposals and then do the classification and localization on these region proposals separately, YOLO is a one-stage object detection network. It divides the image into grids and directly does the bounding boxes regression and classification on these grids. YOLO is speedy and achieves 45 frames per second without sacrificing much accuracy on object classification. However, compared with the object localization, more errors occurred than that in region-based detectors. After that, they continue to propose several different visions of YOLO, such as YOLOv2 (J. Redmon & A. Farhadi, 2017), YOLOv3 (J. Redmon & A. Farhadi, 2018) to improve the accuracy and efficiency.

To achieve real-time speed and without losing too much accuracy, SSD (W. Liu et al., 2016) was designed to eliminate the computation of Region Proposal Network. By applying several convolution filters on multi-scale features, SSD detects the objects directly on the convolutional map. With the idea of using higher resolution feature maps to detect small objects and lower resolution feature maps to detect large objects, the accuracy is significantly improved compared to the former one-stage detector YOLO, which simply detects the objects on the top layer of the convolutional feature maps.

The main reason for the one-stage detectors' performance on accuracy inferior to the two-stage detectors is lacking the region proposal network to balance the positive and negative training samples (T. Lin et al., 2017). In the one-stage detectors, most of the boxes are the background classes that are not useful to training the network, while in the

two-stage detectors, with the help of RPN, the foreground and background samples ratio is fixed with 1:3. Instead of adding more complex network architecture, Lin et al. propose a new loss function called Focal Loss to figure out the imbalanced sample problem. The gist is that: in order to train the rare classes better, they down-weight the easy examples and pay more attention to training the hard-negative examples. Focal Loss makes the one-stage detectors achieve better accuracy than two-stage detectors like Faster RCNN and keep the high detecting speed simultaneously.

2.3 Object detection with Graph reasoning

As object detection systems have been reviewed, it can be noticed that they all treated object detection as a perception problem, not a reasoning problem, and all the region-based object detection systems tried to classify the region of proposals in parallel. In other words, the problem of detection was treated as a simple region classification with a deep convolutional neural network but without considering objects-to-objects relationships or instance-level context. Such strong semantic information should be helpful to the current object detection system. Spatial Memory Network (SMN) was presented (X. Chen et al., 2017) to fill the gap by modeling instance-level context to help object detection reasoning. They created a spatial memory that is used to store previously detected objects and feed them into another convolutional neural network for object-to-object context reasoning to help the detection of the next region proposal. This process keeps going on until all the iterations have been reached. Essentially the spatial memory builds a visual knowledge that provides spatial and semantic interactions between objects. However, this approach has three shortcomings: (1) They use a stack of convolutions to perform local pixel-level reasoning, which lacks scene-level context. (2) They do an iteratively reasoning process based on the previous detection, which is slow and highly dependent on previous

detection results. (3) Their inter-region relationships are implicit, so their improvement is limited.

Not only focus on the objects-to-objects relationships at the instance level but Y. Liu et al. (2018) also incorporated scene-level contexts to help the reasoning process. In their Structure Inference network (SIN), the object prediction is not only decided by its visual appearance but also influenced by the objects-to-objects relationships and scene contexts. They formulated the structure inference as a reasoning problem in a graph where nodes denote the objects, the edges denote relationships between the objects, and these objects interact via the graph under the guidance of scene contexts. The scene context is the whole image visual feature extracted through the same layer as for nodes. As for the edges, they combined the spatial relationships and scene contexts through two learnable weights. Through iteratively updating, the final integrated node representations are used to predict object categories and bounding box offsets.

H. Hu et al. (2018) firstly introduced a fully end-to-end relation network for object detector. The goal of their model is to combine the information of all objects to improve the accuracy of recognition of each object, and the information from other objects is represented by feature vectors which are produced by each relation module. Each relation module uses the geometry and appearance features of all objects as input. After obtaining different relation information between each pair of objects, the new object features, which are reweighted by this relation information, will be concatenated with the original features as the final feature map and fed to the classifier of the object detector. Although they proved that the relation module learned some information between objects that could help current object detection, it is not clear what is learned in the relation module. Moreover, the reasoning process is still based on the convolutional feature map, and the relation between each category is implicit.

In order not to focus on utilizing the convolutional feature of an image itself, but to incorporate external knowledge into the object detection system, Y. Fang et al. (2017) started to integrate the knowledge graph into the existing object detection models. They used the knowledge graph and probability matrix which is the output of initial object detection, to produce a new probability matrix. It means that they used prior external knowledge to reweight the softmax probability distribution.

They designed two different kinds of knowledge graphs: One is based on the frequency of co-occurrence for each pair of categories in the background dataset; another one is based on the relationship provided by the large-scale dataset. The knowledge graphs were modeled and quantified by the symmetric matrix to represent the numerical degree of semantic consistency for each category. With this ideal, when the semantic consistency of two categories is large in the knowledge graph, the probability of two categories should be similar in the same image. Through the re-optimization process, they incorporated knowledge graphs into current object detection models and significantly boosted the performance of baseline networks.

X. Chen et al. (2018) proposed an Iterative Visual Reasoning Beyond Convolutions approach, which has two modes: One local model inspired by the spatial memory method (X. Chen & A. Gupta, 2017) stores previous beliefs with parallel updates, focusing on the reasoning within the convolution. Another global model reasoning beyond convolution through a fixed global knowledge graph containing spatial and semantic relationships allows regions to directly communicate information with each other and the true global relationship between classes. The final prediction combines the results of two models with an attention mechanism, and the accuracy was improved significantly. However, they still propagate the region-wise feature in one image to reason in the local model and with the fixed knowledge to reason in the global model. If the feature representations of

these region proposals are not good enough, due to the problems of heavy occlusion and class ambiguities that are very common in large-scale detection, their reasoning accuracy will be affected heavily.

In the notice of this problem, H. Xu et al. (2019) find out a global reasoning method. Instead of propagating the region-wise feature presentations through the graph constructed from the convolution within the image, they choose to propagate the external knowledge graph's semantic information to a global semantic pool. The global semantic pool represents all the classes. The external knowledge graph provides fully connected edges between each pair of classes in the global semantic pool. After the reasoning process, the new global semantic pool feature representation is enhanced. By mapping the enhanced global semantic pool to region proposals, the region proposals feature representation will also get enhanced, and the final prediction on these enhanced region features will get better accuracy. The method of incorporating prior knowledge is through concatenating the external feature vectors with original region feature vectors. Their global reasoning process is based on the region features simultaneously, so that it can avoid doing an iterative reasoning process on one region proposal each time. However, considering that the human knowledge base is strong and complicated, the knowledge base for object detection is constructed by only one factor: an attribute or co-occurrence frequency. It will be not general or robust enough for all the classes. The semantic gap between the knowledge base and individual image will make the reasoning process less accurate.

2.4 Few/zero-shot learning

The success of the current deep learning model is dependent on a large dataset. In the case of insufficient data, especially in zero-sample learning, knowledge must come in

handy. The main idea of few/zero-shot learning is to learn some new objects with a few samples or zero samples. It utilizes the potential semantic relationship between samples so that the model can process some samples that have never been processed before. The fundamental idea of few/zero-shot learning has similarities with this research, so few/zero-shot learning will be briefly reviewed in this chapter.

The framework of few/zero-shot learning can be mainly divided into three parts:(1) Learning the feature space X of samples, such as extracting image features by deep neural networks; (2) The description of the class in the semantic space A, that is the construction of the potential semantic relationship between the seen classes and the unseen classes; (3) The mapping between feature space X and semantic space A. As CNN could easily extract feature representation, the most important thing is to build a semantic space A. For most of the early works, researchers try to build a latent semantic space before mapping image features to this space by a leverage attribute description and semantic embedding. For example, C. Lampert et al. (2009) firstly constructed the feature representation space X of the samples through the predefined features. Then, through several class collections, it learns attribute descriptions that can be used to represent all classes in the dataset to build semantic space A. Finally, they proposed two ways to establish the mapping between X and A. The two methods are Direct Attribute Prediction and Indirect Attribute Prediction. Although the performance is not as good as supervised deep learning methods, it does express the idea of "knowledge transfer" to a certain extent.

Furthermore, A. Frome et al. (2013) proposed a deep visual-semantic embedding model called DeViSE, which leverages semantic information distilled from the unannotated text to learn semantic relationships between image labels, and maps images to this semantic embedding space. They used the pre-trained skip-gram model to obtain the fixed-length embedding vectors to represent each term, and a pre-trained CNN-based

model to extract visual feature vectors. Then, two vectors are mapped to the same dimension of space and calculate the similarity. During the test stage, images can be classified in line with the similarity in the embedding space.

The knowledge graph that carries rich semantic information has become a natural help in establishing semantic relationships with zero-sample learning. In 2018, X. Wang et al. proposed using semantic embedding and categories relationships to help the classifier. The model is divided into two independent parts. First, a CNN is used to extract image feature vectors. The second part is graph convolutional network. Each class is in the form of semantic embedding and taken as a node of the knowledge graph. After a series of graph convolution operations, it learns a set of weights to represent each class. In the training process, the visual classifier with a small part of classes is used to learn GCN parameters. In the testing phase, the visual classifiers will be used to predict unseen classes.

It is believed that with the development of deep learning, more and more deep learning models will incorporate prior knowledge and reasoning ability.

CHAPTER 3: EXPERIMENTAL METHODS

As the research object in this research is to improve current graph reasoning in Reasoning RCNN. In order to reduce the semantic gap and to make knowledge graph reasoning more accurate and robust, a single knowledge graph will be extended to multiple knowledge graphs with learning parameters to control the weights of each knowledge graph. By fusing more single factor knowledge graphs, each pair of the classes' final relation will be more robust and general.

In this chapter, the whole system architecture will be introduced first in chapter 3.1. Afterward, some essential concepts in this system will be explained in detail, such as how to build a global semantic pool will be presented in chapter 3.2; multiple knowledge graphs convolutional network will be set up in chapter 3.3; attention and mapping classes to region proposals mechanism will be explained in chapter 3.4.

3.1 Overview

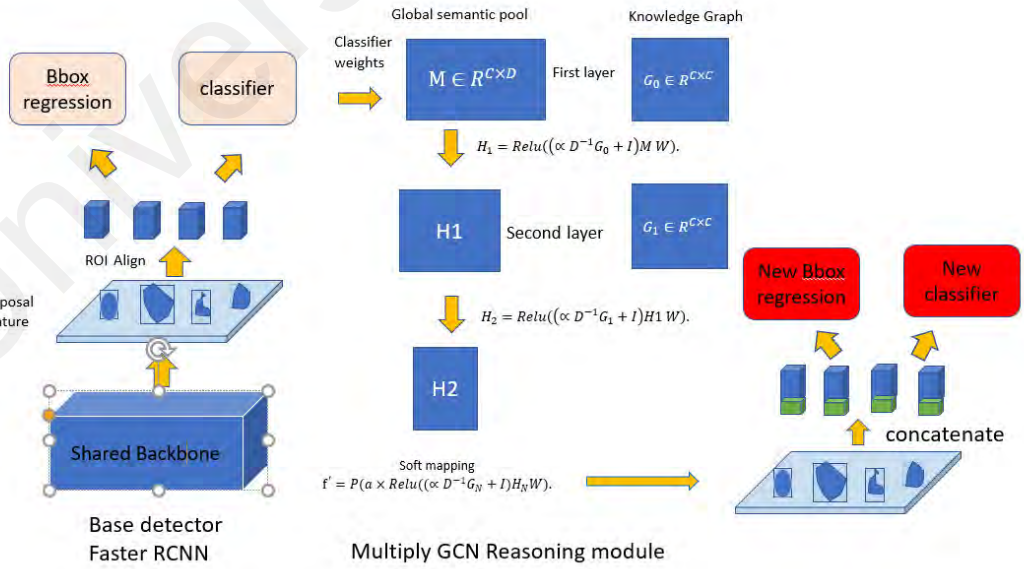

Figure 3.1: The architecture of the system

Figure 3.1 shows the overview of our framework. This system could be divided into two parts or two stages. The first stage on the left of the figure is the baseline of the framework, which is the state-of-the-art object detector Faster-RCNN with FPN. Pretrained ResNet-50 on ImageNet is taken as the shared backbone network to compute the convolutional feature map of images. The second stage, which is in the right, is the multiple graphs convolutional networks (GCN) reasoning module, which is built on the top of the first stage object detector head.

Firstly, a global semantic pool M was built by copying the parameters of the first stage classifier weights to represent semantic information of all the categories. Through the different relationships provided by the multiple knowledge graphs, the final relations between each pair of classes will be more general and robust, and each category's semantic information will be stronger and richer after the feature aggregation process by graph reasoning. As it is known that the reasoning process is global wise, and not all categories exist in one image, a which is squeeze-and-excitation (J. Hu et al., 2018) attention mechanism is used to emphasize the relative categories and suppress the irrelevant categories. After graph convolutional network forward procession, the new global semantic pool will be mapped back to the region proposals. In the last, the enhanced region proposals feature f' will be concatenated to the original region proposals feature f to be F ($[f': f]$), and F will be fed into new bounding box regression and new classifier to get better results.

3.2 Global Semantic Pool M

Because the current object detection problem is modeled as a classification problem on each region proposal feature, and these region proposal feature representations directly affect the accuracy of the object detector. Furthermore, most of the former works (X.

Chen et al., 2017; Y. Liu et al., 2018; Y. Fang et al., 2017; H. Hu et al., 2018) incorporating the knowledge to reasoning process were based on the region proposal features. Their knowledge provided the geometry and semantic relations between each region nodes. However, when in the complex scene, extracting feature representation from the convolutional neural network is affected by several common problems, such as heavy occlusions or overlap, class ambiguities and tiny objects, etc. These problems will ultimately reduce the final classification accuracy. In order to avoid this problem, the highly semantic representation of the whole classes, like the objects' appearance store in human's memory, would be propagated through the knowledge graphs instead of using region feature extracted by the backbone convolutional neural network.

When coming to build a global semantic pool to store the semantic representations of all classes, there are different ways to do that. The first one is using the clustering method. One class feature representation can be obtained from taking the mean average of all the same objects' feature vectors in a specific dataset, but this method is too expansive to compute and not able to train an end-to-end way. Another easy way is to leverage the classifier weight vectors to represent a specific category since it describes which class the object's feature belongs to.

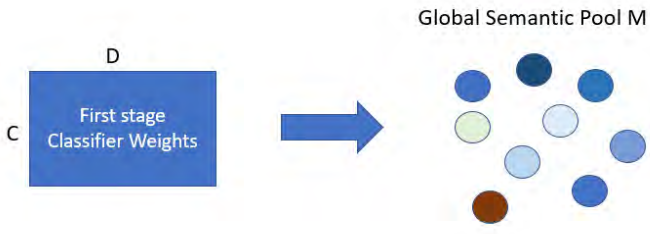

Figure 3.2: Global Semantic Pool M with C categories and D dimensions

So, the global semantic pool M ∈ $R^{C\times D}$ was built from the first stage Faster-RCNN classifier weights, which represent the whole C categories. The global semantic pool can highly represent the semantic information for each category, because it records the feature activations that are trained from all the images, and it will become more and more accurate with the end-to-end training procedure.

3.3 Multiple Knowledge Graph Convolution Networks

Graph convolution networks (GCN) (T. N. Kipf & M. Welling, 2017) is one type of neural network, which is used for machine learning directly on graphs. GCN can leverage structural information from knowledge graph to produce useful feature representation of nodes in graphs. With a given graph $G = graph\,(V, E)$, an $N \times N$ matrix representation of the graph structure such as the adjacency matrix and a feature matrix with the dimension of $N \times D$ are taken as the input of the GCN, where N represents the number of nodes and D represent the dimensions of the feature. The hidden layer in the GCN could be represented as $H_{i+1} = f(H_i, G)$, where the H_i is a feature matrix of the previous layer with $N \times D$ dimension; each row of the matrix represents a node. f represents the propagation rule. In each layer, these features will be aggregated with given structure information from each graph to generate a new feature representation. In the Last, the GCN's output will encode prior knowledge from the graph into the nodes feature representation.

However, the early work Reasoning RCNN only used one handcraft knowledge graph. The big semantic gap between an external linguistic statistic and an individual image makes one single factor handcraft knowledge graph lack universality and robustness. To solve this problem, one possible way to do that is fusing multiple handcraft knowledge graphs which were built from a single factor. By adding more hidden layers in the GCN

with more different knowledge graphs, the more complex and general knowledge will be learned. In the last, each node will be aggregated with more relevant neighbor nodes. In this research, global semantic pool M will be taken to propagate through multiple knowledge graphs G to get highly aggregated feature representation H which can be defined as:

$$H_{i+1} = Relu\big((\propto D^{-1}G_i + I)H_iW\big)$$

Instead of one graph convolution layer with a single knowledge graph, N graph convolution layers with N knowledge graphs are taken in this research as the graph convolutional network. In the first layer, the global semantic pool M is the input H_0 , and G_0 is the first knowledge graph, D^{-1} is the normalization matrix, I is the identity matrix to remain the feature of each own category. $\propto$ is the learning parameter which is to control how many ratios of this knowledge graph will be used. If $\propto = 0$, then it means the current categories have no relation with their neighbors, and thus they do not need neighbors' information to help the feature enhancement, and the current knowledge graph will be not used. Therefore, the learning parameter $\propto$ will identify how related the current knowledge graph to the universal relationship between each category.

$W \in R^{D\times E}$ is the transformation weight matrix which transforms the dimension of input, from D dimension to E dimension. Relu activation function is added to make it nonlinear. In the second layer, the output of the first layer H_1 is taken as the input of the second layer, the second knowledge graph G_1 is build in which is to learn a different relationship between each pair of categories. With more accurate and robust knowledge graphs, the final relationship between each pair of categories will be more general, the gap between the external linguistic information and the individual image will be smaller. Our reasoning module shows in Figure 3.3.

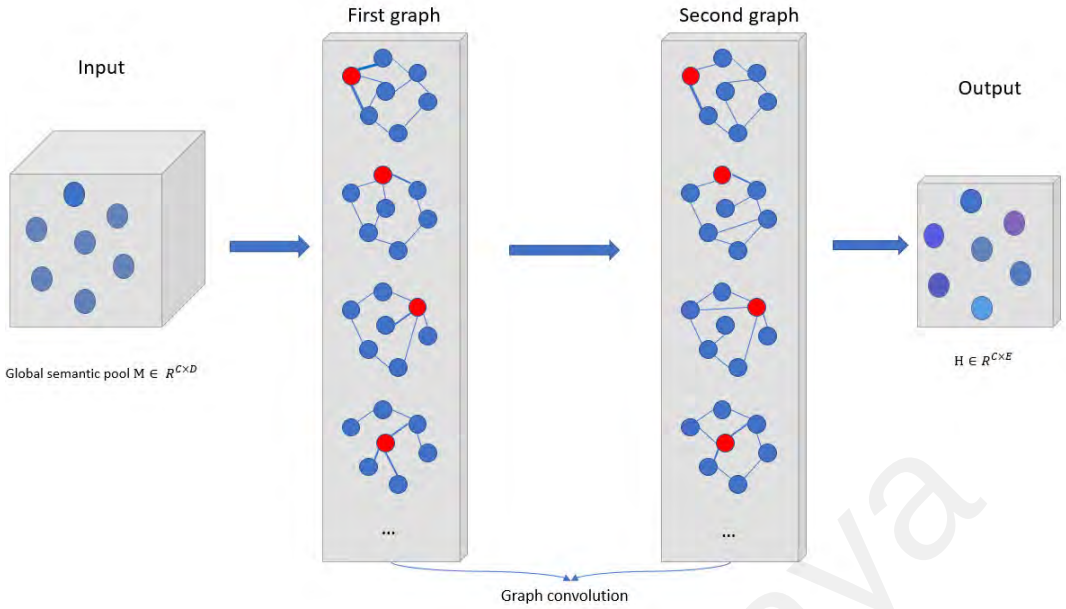

Figure 3.3: Graph convolutional network with multiple knowledge graphs

3.4 Category-wise Attention and Mapping Back to Region Proposals

The output of the multiple knowledge graphs reasoning is feature representations of whole class nodes. The next step is to map all classes feature representation back to the region proposal nodes. As not every class exists in one image, the relative classes need to emphasize that the irrelative classes should be suppressed. So, an attention mechanism needs to add here to make the system focus more on the relative classes. In this research, Squeeze-and-Excitation Network (SENet) (J. Hu et al., 2018) will be modified and used.

For the CNN network, its core calculation is the convolution operator, which learns a new feature map through different convolution kernels. Essentially, convolution is the feature fusion of local area pixels, which includes spatial and inter-channel feature fusion. For convolution operations, a large part of the work is to improve the receptive field, which is to fuse more features spatially or to extract multi-scale spatial information, such as the multi-branch structure of the Inception network (C. Szegedy et al., 2015). For the channel-wise features fusion, the convolution operation defaults to a fusion of all channels

of the input feature map. The innovation of the SENet is to focus on the relationship between channels, hoping that the model can automatically learn the different importance of channel features. To this end, SENet proposed the Squeeze-and-Excitation (SE) module.

The SE module first performs a Squeeze operation on the feature map to obtain channel-level global features. Then it performs an Excitation operation on the global features to learn the relationship between each channel, and also obtain the weights of different channels. Finally, it multiplies the original feature map to get the final characteristics. Essentially, the SE module performs attention or gating operations in the channel-wise dimensions. This attention mechanism allows the model to pay more attention to those channels which have richer information and suppressing those unimportant channel features. Another point is that the SE module is universal, which means it can be embedded into the existing network architecture.

In this research, the squeeze module was designed by one convolutional layer and a global average pooling layer. Firstly, the whole image feature map with the dimension of $H \times W \times D$ is extracted from the shared backbone ResNet. Then it will be put into one convolutional layer to compute the spatial-wise and channel-wise feature relationship, and the number of the convolutional kernels is the same as image feature dimension D. In order to learn the channel-wise relationship other than a spatial-wise relationship, the global average pooling layer was designed to encode spatial feature map $H \times W$ as $\frac{1}{H \times W}$ to represent the global feature descriptor of this channel. Till this end, the squeeze operation is done. The next step is excitation operation,

The squeeze operation obtains the characteristics of global description, and we need another process to capture the relationship between channels. This operation needs to meet two criteria: First, it must be flexible, it must be able to learn the nonlinear

relationship between each channel; the second point is that the learned relationship is not mutually exclusive because multi-channel features are allowed here instead of one-hot form. Based on this, the gating mechanism in the form of the sigmoid is used here. To reduce the model complexity, and to improve the generalization ability, a bottleneck structure including two fully connected layers is adopted here. The first FC layer reduces the dimensionality reduction effect. The dimensionality reduction coefficient is a hyperparameter, and the second FC layer restores its original size. Finally, the activation value of each channel $1 \times D$ will be learned.

After acquiring the channel-wise attention descriptor, the related categories attention $1 \times C$ can be computed in a subtle way which is to associate channel-wise attention descriptor to the global semantic pool. It should be noted that the dimension of the channel in the global semantic pool $\mathrm{M} \in R^{C \times D}$ is also D. Through matrix multiplication with global semantic pool transpose $M^T \in R^{D \times C}$ and softmax activation function, the new attention scale $a \in R^{1 \times C}$ can be got to represent the weights of importance of each class. Once the categories attention descriptor was obtained, the relativities of each class will be rescaled by multiplication with the whole categories feature $H_i \in R^{C \times E}$. In the last, the drawback of global reasoning, which is the noise of the irrelative categories, is improved by the category-wise adaptive attention mechanism.

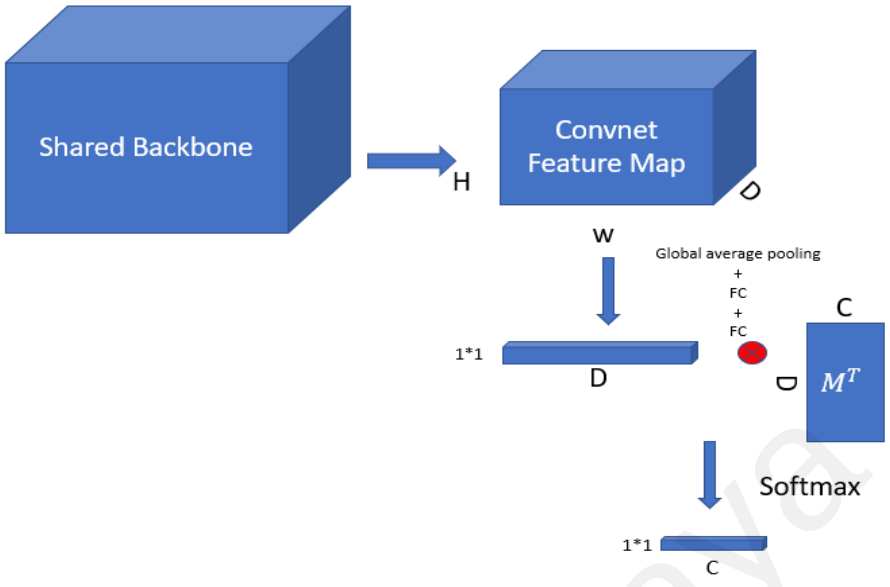

Figure 3.4: Category-wise attention mechanism

Because the current strategy of Faster RCNN is classification on the selected region proposals, and our reasoning method is on the global semantic pool. Hence, the next step is to map the whole C classes semantic pool to Nr region proposals. By matrix multiplication of classification probability distribution P $\in R^{Nr\times C}$ that from the output of the first stage classifier, with the whole categories feature $H_i \in R^{C\times E}$, the global semantic pool is mapped to region proposals $R^{Nr\times E}$ eventually. A new highly semantic feature representation of region proposals $\mathrm{f}' = P(a \times Relu((\propto D^{-1}G_N + I)H_N W)\ ) \in R^{Nr\times E}$ could be got.

3.5 Feature Map fusion

Because the highly semantic feature map f′ is got from the global reasoning, it would contain higher semantic information than the original region proposal feature map f, which is extracted from the first stage convolutional neural network and region proposal

network. How to fuse different level semantic feature is crucial before the feature maps are fed into a new classifier. One natural way to fuse these two feature maps is feature concatenation. A fully connected layer will be added here to learn the weights of each dimension of region proposal features. The original feature f $\in R^{Nr\times D}$ could be got from the ROI pooling layer in the first stage network. After concatenation operation, the dimension of each region proposal feature X_j will become (D+E).

$$\begin{bmatrix} X_1^1, X_1^2, \ldots, & \ldots, X_1^{D+E} \\ X_2^1, X_2^2, \ldots, & \ldots, X_2^{D+E} \\ \vdots & \\ \vdots & \\ X_{Nr}^1, X_{Nr}^2, \ldots, & \ldots, X_{Nr}^{D+E} \end{bmatrix}$$

Figure 3.5: Final feature map for classifier

After the learning process by a fully connected layer, the features of different dimensions will be fused. Finally, it will be fed into a new softmax classifier to calculate the probabilities of different classes.

CHAPTER 4: EXPERIMENT AND RESULTS

4.1 Dataset and Evaluations

The experiments are set on the Google Colab, which provides free GPU and RAM. As the limits of the hardware devices, the dataset should not be too large. Hence, a small commonly used dataset in the object detection area: PASCAL Visual Object Classes (PASCAL VOC) (M. Everingham et al., 2010) is selected for our dataset. The PASCAL VOC challenge is a benchmark for object detection and category recognition. It provides a standard dataset of images and annotation and standard evaluation procedures to the computer vision and machine learning communities. It was arranged once a year since 2005, its related dataset and challenge have become accepted as the benchmark for object detection.

For the training set, the union of PASCAL VOC2007 training set and VOC2012 training set are used, which is about 10K images. For the testing set, the system is evaluated on the VOC2007 test, which is about 4.9K images. As image feature extracted from CNN is not invariant to rotation and scale changes, and bounding boxes existing in object detection system limits the ways of data augmentation technique. In this research, only image re-sizing and image flipping are used as data augmentation to make the object detection system more robust. So, the input images will be resized to the same scale with 1333×800 resolution, and a random flip with a ratio of 0.5 is adopted in both training and testing stages.

When evaluating a deep learning model, the speed and accuracy are always be compared and wished to get the higher place. However, these two important evaluation criteria cannot be achieved at the same time, especially in the object detection domain.

Many practical object detection applications have high requirements for accuracy and speed. If speed performance indicators are not considered, models with higher precision always require higher computational complexity. Generally, the speed evaluation indicator in object detection is FPS, which is the number of pictures that the detector can inference at each one second, or the time that the detector needs to process each picture. But the speed evaluation index must be carried out on the same hardware. Its maximum FLOPS (the float point number of operations per second) represents the hardware performance is the same. For the different networks, the time required to process each picture is various. The speed of the object detection system is affected by many factors, such as the number of layers of your network, parameters, the selected activation function, etc. The lower number of the parameters, the smaller the FPS will be, the model required memory is smaller, and the hardware memory requirements are relatively low.

On the other hand, the object detection system's accuracy indicator: mean average precision (mAP), is a little bit complex. Before talking about the mAP, several concepts need to be mentioned, such as True Positives (TP), False Positives (FP), and False Negatives (FN). For determining the predicted object belongs to which bounding box, the IoU threshold value needs to be set. For example, IoU is set to 0.5. If the predicted bounding box and ground truth box IoU ≥ 0.5, it will be classified as TP. If IoU < 0.5, then it will be classified as FP. If the ground truth box is in the image and the system failed to detect this object, it will be represented as FN. Precision defined as $\frac{TP}{TP+FP}$ and Recall defined as $\frac{TP}{TP+FN}$ are the two important metrics used to evaluate the performance of object detectors. There is always a trade-off between Precision and Recall. Increasing one of them usually decreases the other one. Sometimes Precision-Recall (PR) graph is not always monotonically decreasing due to certain exceptions.

Since the mAP is the mean value of AP of all categories in the dataset, it is important to know how to calculate a certain category's AP value. The AP calculation methods of a certain category of different datasets are similar, mainly divided into three types: (1) Before VOC2010, you only need to select the maximum Precision when Recall ≥ 0, 0.1, 0.2, ..., 1, with a total of 11 points. AP is the average of these 11 Precisions. (2) After VOC2010, for each different Recall value (including 0 and 1), select the maximum Precision when it is greater than or equal to these Recall values, and then calculate the area under the PR curve as the AP value (3) COCO dataset, set multiple IoU thresholds (0.5-0.95, 0.05 is the step size), each IoU threshold has a certain category of AP value, and then calculate the average AP value under different IoU thresholds, that is the final AP value of a certain category. Generally, the mAP is for the entire dataset, and it evaluates the overall performance of the object detection system; AP is for a certain category in the dataset, it evaluates the system performance on a specific category.

In this research, the accuracy of the system is only evaluated, and the second evaluation method with IoU thresholds (0.5) is applied as a standard metric to compare the system performance with other networks.

4.2 Knowledge Graph

The current knowledge graph is obtained through statistics of the VG dataset's annotation by H. Xu et al. (2018). They build two knowledge graphs which are attribute graph and co-occurrence graph. Specifically, the attribute graph was constructed by defining the attribute similarities to be the edges of each pair of classes. They considered the most 200 frequent attribute annotations in the VG dataset such as color, materials, and size, etc. The attribute probability distributions were calculated by frequency statistics for each class. The similarity E_{ji} between class C_j and class C_i was calculated by the attribute

probability distributions $P_j \; and \; P_i$ though Jensen Shannon divergence (JS divergence) $E_{ji} = JS(P_j||P_i)$. The similarity E_{ji} will be the edge between class C_j and class C_i in the attribute graph.

The co-occurrence frequency graph was constructed by the statistics of the pair-wise classes co-occurrence in the VG dataset. They first counted the number N_{ij} of class C_i and class C_j when they are co-occurrence, after row and column normalization $E_{ij} = \frac{N_{ij}}{\sqrt{D_i * D_j}}$, where $D_i = \sum_0^j N_{ij}$, $D_j = \sum_0^i N_{ij}$, E_{ij} becomes the edge of each pair of classes.

Because VG dataset is a large-scale dataset that contains 3000 classes, it needs to be made adapted to the different datasets such as MSCOCO (80 classes), PASCAL VOC (20 classes), etc. Edges are distilled from the relevant classes which are in the target datasets, and new knowledge graphs will be suitable for different datasets. Some parts of knowledge graphs that fit the PASCAL VOC are showing in Figure 4.1.

	Aero plane	bicycle	bird	boat	bottle	bus	car
Aero plane	**0**	0.88	0.85	0.96	0.78	0.95	0.91
bicycle	0.88	0	0.86	0.85	0.80	0.90	0.94
bird	0.85	0.86	0	0.85	0.79	0.81	0.86
boat	0.96	0.85	0.85	0	0.81	0.94	0.86
bottle	0.78	0.80	0.79	0.81	0	0.82	0.76
bus	0.95	0.90	0.81	0.94	0.82	0	0.87
car	0.91	0.94	0.86	0.86	0.76	0.87	0

Attribute graph

	Aero plane	bicycle	bird	boat	bottle	bus	car
Aero plane	0	0	0.019	0.001	0	0.009	0.002
bicycle	0	0	0.003	0.013	0.03	0.085	0.068
bird	0.019	0.003	0	0.134	0	0.001	0.039
boat	0.0012	0.012	0.134	0	0.001	0.003	0.01
bottle	0	0.03	0	0.001	0	0.002	0
bus	0..009	0.085	0.001	0.003	0.002	0	0.2
car	0.027	0.068	0.039	0.01	0	0.2	0

Co-occurrence graph

Figure 4.1: Visualization for parts of the Attribute graph and Co-occurrence graph

As it can be seen, from these two different knowledge graphs, the edge weights between different classes vary largely due to the different graph construction methods. In order to reduce the effect of human subjective factors and make the relation between

classes more accurate, the reasoning module is modified from one single knowledge graph to multiple knowledge graphs. In this research, both knowledge graphs are used in the reasoning module.

4.3 Implementation detail

Faster-RCNN with FPN is treated as a baseline in this framework, pre-trained ResNet-50 (K. He et al., 2016) on ImageNet (O. Russakovsky et al., 2015) as the backbone network, and it will be frozen when training the whole network. The module of multiple knowledge graphs reasoning is built on top of the backbone network as the second part. For the RPN network, 2000 region proposals are assigned firstly. Those proposals having IoU > 0.7 with ground truth boxes will be labeled as positive proposals, and IoU < 0.3 with ground truth boxes will be labeled as negative proposals. After applying the Non-Maximum Suppression (NMS) algorithm, 512 region proposals are sampled. After ROI pooling, these region proposal features are fed into two Fully connected layers and then become the original feature f (dimension = 1024), which is the input of the classifier and bounding box regression. With the whole training forward process in the Faster RCNN, the first stage network is done.

The second stage is the reasoning module. The same RPN network is used to extract the region features. After that, the global semantic pool M $\in R^{C\times D}$ $(C = 20, D = 1024)$ was built by copying the parameters of the classifier weights of the first stage network Faster RCNN. It then will be fed into the first layer of the reasoning module.

The reasoning module takes the ideal of the graph convolution network, which is intended to take the connections of the edges to aggregate the original nodes' features and output new nodes' features. In this paper, the feature matrix M is taken as our nodes'

features input, and ($\propto D^{-1}G_0 + I$) as our edges information. The identity matrix I is to preserve the original features as much as possible and avoid too many features of neighbors occupying in the aggregation process. The hyperparameter $\propto$ is used to manually control how many ratios of the neighbors' feature will be incorporated. In order to reduce the computation cost, the dimension of the first layer's output is transformed from 1024 to 256. In the second layer, the output of the first layer is propagated through the second graph to get a new whole categories' feature representation with the same ideal of the first layer but keep the same dimension.

After aggregating related nodes' feature according to the relation of knowledge graphs, the new global semantic pool H $\in R^{C\times E}$ ($E = 256$) is needed to map back with region proposals. So the classes probability distributions P $\in R^{Nr\times C}$ (Nr = 512) for all the regional nodes are taken from the output of the first stage classifier, to multiply classes feature representation H $\in R^{C\times E}$ to get the enhanced feature representation of region nodes $f' \in R^{Nr\times E}$. The enhanced region nodes feature vector f' will be concatenated with the original region proposal f $\in R^{Nr\times D}$ which is the output of the RPN in the first stage network, and final region proposals feature F $\in R^{Nr\times(D+E)}$ will be fed to train a new classifier and bounding box regression.

In the backward step, Cross-Entropy Loss is applied as the loss function for the classifier in the RPN and detector heads. Smooth L1 Loss will be calculated for all the bounding box regressions. To make the new classifier and bounding box regression in the second stage to learn better and faster, the weights of loss were adjusted, where the loss in the first stage detector head was reduced to half. For the optimizer, Stochastic gradient descent (SGD) is adopted as our optimizer strategy. The initial learning rate Lr $= 0.01, \text{momentum} = 0.9, \text{and a weight decay rate} = 0.0001$ are set to optimize all the networks. After nine epochs, it will decay the initial learning rate.

At the testing period, the RPN network will generate 1000 region proposals with an NMS threshold of 0.7. At the same time, the RCNN network will predict based on the 1000 region proposals with IoU $>$ 0.5.

4.4 Influence of Parameter $\propto$

Three experiments are set firstly to test if the training parameter $\propto$ is helpful. With all the same settings except three sets of two fixed parameters $\propto$ for two graphs respectively, after training six epochs, the mean average precision of our system can be obtained. The following Table 4.1 shows the results.

Table 4.1: System performance with fixed parameters for two knowledge graphs

Parameter $\propto$ for co-occurrence frequency graph and attribute graph	mAP
1.0 / 0.25	0.489
1.0 / 0.5	0.524
0.8 / 0.6	0.480

It can be found out that by only changing the parameter $\propto$, the performance of the whole system changed largely. In order to find the optimized parameter $\propto$ and make our system performance better, the parameter $\propto$ is made to be learnable. After training 12 epochs, the final mean average precision is 0.702, and the parameters are 2.6 and 0.8 for each knowledge graph.

Table 4.2: The optimum system performance

Class	AP
Aero plane	0.777
bicycle	0.778
bird	0.621
boat	0.615
bottle	0.61
bus	0.793
car	0.863
cat	0.756
chair	0.525
cow	0.695
Dining table	0.63
dog	0.682
horse	0.798
motorbike	0.784
person	0.814
Potted plant	0.429
sheep	0.699
sofa	0.693
train	0.803
Tv monitor	0.676
Mean AP	**0.702**

From Table 4.2, it can be found out that some classes' performance is poor such as potted plant, bird, bottle, which are small objects. In contrast, the classes like person, car and train which are bigger objects, their average precision are higher. This phenomenon is very common in object detection due to the imbalanced training example (the sample size of some classes is small) and tiny object feature representation. In this research, the aim is not to work out the poor precision on small objects but to improve the current reasoning module in feature enhancement so that it can improve the overall accuracy of the object detection system.

4.5 Comparison with early work

To compare with the Reasoning RCNN, which reasoning through a single knowledge graph, the code of Reasoning RCNN with the same baseline (Faster RCNN with FPN) is implemented. The only difference is the reasoning module, in which they use a single

knowledge graph. In contrast, multiple knowledge graphs are applied with the learning parameter to control how much each knowledge graph is used. The system without reasoning module, which is plain Faster RCNN is also evaluated. After training 12 epochs on PASCAL VOC2007 and PASCAL VOC2012 dataset, each system performance is evaluated by mean average precision as showing in Table 4.3.

Table 4.3: Map comparison

Methods	Backbone	Knowledge Graph	map
Faster RCNN	Resnet-50-FPN	No reasoning module	0.642
Reasoning RCNN	ResNet-50-FPN	Attribute Graph	0.650
Reasoning RCNN	ResNet-50-FPN	Co-occurrence Graph	0.674
Our Method	**ResNet-50-FPN**	**Co-occurrence Graph + Attribute Graph**	**0.702**

From Table 4.3, It can be found out that the performance of Reasoning RCNN is affected by the different external knowledge graphs. Compared with Faster RCNN, Reasoning RCNN with attribute graph performance boosted only 1%, Reasoning RCNN with co-occurrence graph performance boosted around 3%. While our method uses both knowledge graphs with learning parameters, the performance is boosted by around 6%. Above all, it can be concluded that the method proposed by this paper can achieve higher accuracy than the Reasoning RCNN and Faster RCNN.

In order to find out the detail which class prediction was improved, the average precision (AP) of each class is also compared among Reasoning RCNN with attribute graph, Reasoning RCNN with co-occurrence graph, and our method. Three models are listed with average precision in each class. As shown in Figure 4.2, the Y-axis represents the average precision value, and the X-axis lists all the 20 classes in the PASCAL VOC. In each class, all three models' average precision values are listed respectively. The blue

color represents the Reasoning RCNN with co-occurrence graph, and the orange is the value of Reasoning RCNN with attribute graph, our method represented by grey color. From Table 4, It can be found out that in most classes, our approach gives the best results, and Reasoning RCNN with attribute graph performs worst.

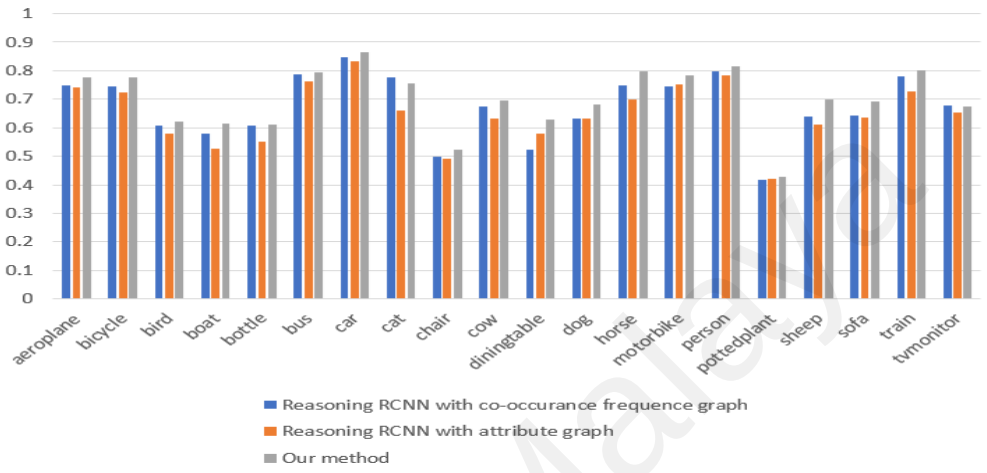

Figure 4.2: Comparison of the average precision on each class

From Table 4.3 and Figure 4.2, by comparing the mean average precision and average precision of each class, it can be found out that not only the overall performance is improved but also the precision of each category.

4.6 Analysis of the improvement

As it has been proved that our system boosted the accuracy of classification, but when having a close look at the essential mechanism of our system, what is the reason that our model improved? The improvement of our system boosted by the reasoning module is feature enhancement which happened before region features are fed into classifier and bounding box regression. This system is not a novel network architecture but stacked on the top of the Faster RCNN. The original region proposal features f $\in$ $R^{Nr\times D}$ where Nr is the number of region proposals, and D is the dimension of each region feature. With

the reasoning module features f ∈ $R^{Nr\times E}$, where E is the dimension of reasoning module output, the region proposal features are enhanced by concatenation operation, which is f′ ∈ $R^{Nr\times(E+D)}$. More accurate classification could be obtained based on the enhanced feature f' when some of the region proposal features were bad due to some reasons such as occlusion. Based on this, what we do is to improve the feature representation through the reasoning module. It is believed that with a more strong and robust knowledge base, the feature out of the reasoning module will be more helpful. However, the current knowledge base in the Reasoning RCNN is built by the statics of a single factor, which is not reliable enough. For this reason, the knowledge base is expanded from a single knowledge graph to multiple knowledge graphs.

CHAPTER 5: CONCLUSION

5.1 Summary

In this research, the current mainstream object detection strategy is studied. It was modeled as parallel image classification without considering the relationship between each object. Such strong semantic information should be helpful to improve and design a better object detection system. Then, the human visual recognition system was analyzed that humans not only rely on the visual appearance of the object but also on the reasoning ability and knowledge base. It takes an important role in the human optical system. After that, the question of how to make an object detection model incorporate the knowledge and reasoning ability was explored, in which knowledge was encoded as a graph and reasoning process was realized by graph convolutional network. Along this direction, the early related works were analyzed, and some drawbacks were proposed in chapter 2. Due to the limitation of region proposals' convolutional feature representations, and relationships within a convolutional neural network are inexplicit, the global wise reasoning is extended as a new research direction. An abstract but with high semantic information global semantic pool was built to replace the convolutional feature map to represent all the categories. The external knowledge base, instead of the relationship distilled from the image convolutional map itself, was incorporated. However, the handcrafted knowledge based on one single factor still cannot be compared with the human knowledge base, which is much more complex and general.

Noticing this problem, more single factor knowledge graphs were incorporated in the global wise reasoning process with learning parameters to balance each knowledge graph's weights. Hence, a novel method, adaptive global reasoning with multiple knowledge graphs for object detection, is presented.

The reasoning module in this system is flexible, it could be embedded in any region-based object detection system, and there is no need to change the base object detector. By adding extra convolutional layers for incorporating more different knowledge graphs and without sacrificing too much computation, Figure 4.3 shows that the accuracy of adaptive global reasoning with multiple knowledge graphs is better than the global reasoning process with a single knowledge graph. Thus, the final relationship between each object should be more general and robust, and the semantic gap between external knowledge with individual images was reduced. The goal of this research was achieved.

5.2 Limitation

As knowledge graph construction is very complex, and collecting data, building knowledge graphs by an individual can cost a lot of computation and time. Thus, in this research, the number of knowledge graphs is limited, and only two knowledge graphs built by the early work Reasoning RCNN were incorporated. To further verify this research's hypothesis, more knowledge graphs should be incorporated to find out if the performance of the whole system will reduce or go to bottleneck when more graph convolutional network layers are added. Furthermore, the knowledge graph construction method should have further study. Currently, the knowledge graph for object detection is handcrafted based on a particular dataset. Therefore, it is restricted to a particular domain and needed to improve the generalization ability to the other domains. Building a strong and complex knowledge base like human common sense is crucial to computer vision tasks and deep learning areas.

When analyzing the system performance on different categories, some categories such as potted plant and chair still show low accuracy. The improvement in these categories is limited. One possible reason is that training samples are not enough because the way of

incorporating prior knowledge in this research is feature vector concatenation. If the original category feature vector is not suitable, then the enhanced feature vector could still be unsuitable. Thus, the system performance is still limited by the imbalanced training dataset problem.

Due to the high hardware requirements of deep learning, it usually takes a long time to train a model on a large data set. So, the experiment was simplified. The experiment on Google Colab can only use a smaller data set and a smaller deep neural network model. Thus, this research only training the network with resnet50 on the Pascal VOC dataset. In order to compare with other state-of-the-art object detection systems, the system performance should also be evaluated in the larger datasets and deeper networks. On the other hand, using a larger data set can better reflect the deep learning models' generalization ability and avoid over-fitting problems.

5.3 Future work

Following this research, there are several ways to explore further. One of them is the knowledge graph. There are good reasons to speculate that the current reasoning module can be continuously improved by considering the local information within the image, such as the spatial distance between the objects. One way to incorporate spatial information is to make use of the spatial distance between region proposals as the weights to adjust the external knowledge graph. For example, taking the previous stage bounding box predictions as region nodes, the distance between region nodes i and j can be calculated by $D = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$, (x, y) is the center of the region nodes. Then distance would be normalized to the range [0,1]. To avoid the redundant connection between background region samples and computational complexity, the most relevant neighborhoods of each region proposal would be only considered. Finally, region nodes

spatial relationship matric $w_g \in R^{Nr \times Nr}$ will be got. To make it fit the external knowledge graph $G_i \in R^{C \times C}$, region nodes should be mapped to the class nodes by $W_g = P^T w_g P$, $P \in R^{Nr \times C}$. Finally, $G_i * W_g$ will be taken as the final adjusted knowledge graph, which incorporates the external knowledge with local image information. Another way to explore knowledge graphs in the future is to focus on how to build a complex that considers multiple factors. Instead of incorporating more different knowledge graphs, building one knowledge graph like human common sense is much better.

The current mainstream knowledge graph representation still has various problems, such as the inability to describe the semantic relevance between entities and relationships, so that it is hard to handle complex relationships. Secondly, the model will be too complex and inefficient in computation due to the introduction of many parameters. Furthermore, it isn't easy to extend to large-scale knowledge graphs. As noticing these problems, there is still a long way to go to provide better prior knowledge to machine learning or deep learning models.

In the way of encoding knowledge to the current deep neural network, it can also be further explored. As it is found out that using feature concatenation operation to encode the knowledge to the deep neural network, the performance of the whole system is limited somehow because the original feature still dominates, thus how to make knowledge play a more important role in object detection system's prediction should be a very crucial and exciting research area. Leveraging the global reasoning module to reweight exist classification probability distribution seems to be a good entry point. For example, if the relationship provided by the knowledge is strong between two categories, and one category has high confidence in the classification probability distribution, and one has low confidence, then the confidence of the low one can be increased. Another way is to modify the current loss function to be knowledge-aware. Many early works try to improve

the accuracy of object detection systems by changing the loss function such as Focal loss. Making deep neural networks adapt to the hard-training category through knowledge encoding is a very challenging task. Finally, how to leverage knowledge to improve the efficiency of the current deep learning model and keep the accuracy at the same time is also worth studying. The real-time requirement of the object detection system will become higher and higher in the future.

REFERENCES

Bar, M. (2004). Visual objects in context. *Nature Reviews Neuroscience*, *5*(8), 617-629.

Cai, Z., & Vasconcelos, N. (2018). Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6154-6162).

Chen, X., & Gupta, A. (2017). Spatial memory for context reasoning in object detection. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 4086-4096).

Chen, X., Li, L. J., Fei-Fei, L., & Gupta, A. (2018). Iterative visual reasoning beyond convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7239-7248).

Chen, X., Shrivastava, A., & Gupta, A. (2013). Neil: Extracting visual knowledge from web data. *In Proceedings of the IEEE international conference on computer vision* (pp. 1409-1416).

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)* (Vol. 1, pp. 886-893). IEEE.

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, Y., Kuan, K., Lin, J., Tan, C., & Chandrasekhar, V. (2017). Object detection meets knowledge graphs. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence* (pp. 1661-1667).

Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008, June). A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-8). IEEE.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A., & Mikolov, T. (2013). Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems* (pp. 2121-2129).

Gidaris, S., & Komodakis, N. (2018). Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4367-4375).

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 580-587).

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, *37*(9), 1904-1916.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

Hoffman, J., Guadarrama, S., Tzeng, E. S., Hu, R., Donahue, J., Girshick, R., ... & Saenko, K. (2014). LSDA: Large scale detection through adaptation. In *Advances in Neural Information Processing Systems* (pp. 3536-3544).

Hu, H., Gu, J., Zhang, Z., Dai, J., & Wei, Y. (2018). Relation networks for object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3588-3597).

Hu, H., Zhou, G. T., Deng, Z., Liao, Z., & Mori, G. (2016). Learning structured inference neural networks with label relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2960-2968).

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7132-7141).

Jiang, C., Xu, H., Liang, X., & Lin, L. (2018). Hybrid knowledge routed modules for large-scale object detection. In *Advances in Neural Information Processing Systems* (pp. 1552-1563).

Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., ... & Bernstein, M. S. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123(1), 32-73.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).

Lampert, C. H., Nickisch, H., & Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 951-958). IEEE.

Le Cun, Y., Jackel, L. D., Boser, B., Denker, J. S., Graf, H. P., Guyon, I., ... & Hubbard, W. (1989). Handwritten digit recognition: Applications of neural network chips and automatic learning. *IEEE Communications Magazine*, 27(11), 41-46.

LeCun, Y. (1998). The MNIST database of handwritten digits. *http://yann.lecun. com/exdb/mnist/*.

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.

Li, Y., Tarlow, D., Brockschmidt, M., & Zemel, R. (2015). Gated graph sequence neural networks. *arXiv preprint arXiv:1511.05493*.

Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).

Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21-37). Springer, Cham.

Liu, Y., Wang, R., Shan, S., & Chen, X. (2018). Structure inference net: Object detection using scene-level context and instance-level relationships. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6985-6994).

Marino, K., Salakhutdinov, R., & Gupta, A. (2016). The more you know: Using knowledge graphs for image classification. *arXiv preprint arXiv:1612.04844*.

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv*:1804.02767.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (pp. 91-99).

Rothe, R., Guillaumin, M., & Van Gool, L. (2014). Non-maximum suppression for object detection by passing messages between windows. *In Asian conference on computer vision* (pp. 290-306). Springer, Cham.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... & Berg, A. C. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, *115*(3), 211-252.

Sadeghi, F., Kumar Divvala, S. K., & Farhadi, A. (2015). Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1456-1464).

Seo, Y., Defferrard, M., Vandergheynst, P., & Bresson, X. (2018). Structured sequence modeling with graph convolutional recurrent networks. In *International Conference on Neural Information Processing* (pp. 362-373). Springer, Cham.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Teney, D., Liu, L., & van Den Hengel, A. (2017). Graph-structured representations for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9).

Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, *104*(2), 154-171.

Vinyals, O., Toshev, A., Bengio, S., & Erhan, D. (2016). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4), 652-663.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* (Vol. 1, pp. I-I). IEEE.

Wang, X., Ye, Y., & Gupta, A. (2018). Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6857-6866).

Xu, H., Jiang, C., Liang, X., Lin, L., & Li, Z. (2019). Reasoning-RCNN: Unifying Adaptive Global Reasoning into Large-scale Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6419-6428).

Zhang, X., He, L., Chen, K., Luo, Y., Zhou, J., & Wang, F. (2018). Multi-view graph convolutional network and its applications on neuroimage analysis for parkinson's disease. In *AMIA Annual Symposium* Proceedings (Vol. 2018, p. 1147). American Medical Informatics Association.