A HYBRID DEEP CNN MODEL FOR FAST CLASS-INCREMENTAL FOOD CLASSIFICATION

AYMEN TAHER AHMED AL-ASHWAL

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

2019

A HYBRID DEEP CNN MODEL FOR FAST CLASS-INCREMENTAL FOOD CLASSIFICATION

AYMEN TAHER AHMED AL-ASHWAL

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

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

2019

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: (I.C./Passport No.:)

Registration/Matric No.:

Name of Degree:

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

Field of Study:

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:

A HYBRID DEEP CNN MODEL FOR FAST CLASS-INCREMENTAL FOOD CLASSIFICATION

ABSTRACT

Food recognition can help in identifying calories, which is particularly helpful in reducing risks that are related to inaccurate food consumption. Recent works used deep learning classifiers for food recognition, which does not have enough intelligence to update the increasing number of food classes. It requires retraining the model for new classes or using transfer learning. Retraining the model takes an estimated time of 20 to 45 hours depending on the accuracy achieved by that specific model. Inspired by the recent success and high performance of Densely Connected Convolutional Networks (DenseNet), in this study, **a hybrid deep Convolutional Neural Network (CNN) model was introduced**. This model has an optimized DenseNet network for features extraction and Adaptive BAll COver for Classification (ABACOC) as an incremental learning classifier. The method employs the intelligence of deep CNN (DenseNet Model) to extract the features after training the model on a wide range of food categories and images. Features are then enhanced by using Tree-based feature selection to reduce the size of each feature and, therefore, enhance classification performance.

Lastly, the incremental learning algorithm ABACOC is used to classify each feature of food classes. The main contribution of this study is a classification model that can predict new classes and incrementally over time improve the accuracy of different food classes. By evaluating the model on food dataset FOOD101, extracting of features take 80.23 seconds and classifying and training of incremental algorithm take 1253.36 seconds with 77.72% test accuracy. Moreover, adding new classes or new food images features has no significant consequence on the model knowledge. On the contrary, new samples for existing classes improve their overall accuracy. Such results are not close to the state-of-art in food classification. However, further research should be done to accomplish higher results with incremental learning.

Keywords: Food recognition, Deep Convolutional Networks,Incremental learning, Features extraction.

MODEL HIBRID "DEEP CNN" BAGI KELAS-BERTAMBAH UNTUK PENGKELASAN MAKANAN SECARA PANTAS

ABSTRAK

Pengenalpastian makanan dapat membantu dalam mengenal pasti kalori makanan, di mana ia membantu dalam membantu risiko yang berkaitan dengan pengambilan makanan yang tidak tepat. Kajian terkini dijalankan dengan lebih mendalam, di mana ia tidak mencukupi untuk mengemas kini peningkatan klasifikasi makanan. Latihan semula model diperlukan untuk kelas baharu atau menggunakan pemindahan pembelajaran. Latihan semula model mengambil masa kira-kira 20 hingga 45 jam bergantung kepada ketepatan yang dicapai oleh model yang tertentu.

Diilhamkan oleh kejayaan yang terkini dan prestasi tinggi oleh DenseNet, dalam kajian ini, satu model hibrid dalam CNN telah di perkenalkan. Model ini mempunyai rangkaian DenseNet yang dioptimumkan untuk ciri-ciri pengekstrakan dan Cover Protector BAll untuk Klasifikasi (ABACOC) sebagai peningkatan klasifikasi pembelajaran. Kaedah ini menggunakan kecerdasan dalaman CNN (DenseNetModel) untuk mengekstrak ciri-ciri selepas melatih model dengan pelbagai lingkungan kategori makanan dan imej. Ciri-ciri kemudiannya ditambah baik dengan menggunakan pemilihan ciri-ciri berasaskan pokok untuk mengurangkan saiz setiap ciri yang akhirnya meningkatkan prestasi klasifikasi. Akhir sekali, penambahan pembelajaran Algoritma ABACOC digunakan untuk mengklasifikasi setiap ciri kelas makanan.

Klasifikasi model yang dicadangkan boleh meramalkan klasifikasi baru dan secara bertahap meningkatkan ketepatan klasifikasi makanan yang berbeza. Dengan menilai kaedah pada set data makanan yang ditanda aras dengan FOOD101, ketepatan sebanyak 77.72% dengan 80.23 saat, untuk mengekstrak ciri-ciri dan 1253.36 saat bagi mengklasifikasi dan melatih peningkatan algorithm. Selain itu menambah klasifikasi baru atau ciri-ciri imej baru tidak mempunyai kesan yang penting kepada pengetahuan model. Sebaliknya sampel baru untuk klasifikasi sedia ada meningkatkan keseluruhan ketepatannya. Hasilnya tidak dekat dengan teknik canggih dalam klasifikasi makanan. Walau bagaimanapun, kajian lanjut perlu dilakukan untuk mencapai hasil yang lebih tinggi dengan pembelajaran yang lebih baik.

ACKNOWLEDGEMENTS

I would like to take the chance and thank my supervisor, Professor Dr. Loo Chu Kiong for his guidance and advice while I am fulfilling my Masters' degree. His help and supervision directed me in the right path in my Masters' journey and therefore I am thankful.

I would also take the opportunity to thank my colleagues in the research lab. They supported me in different ways and always there when required. I would like to acknowledge the support and assistant from my friend Mustafa. I would also like to mention Ahmed Ibrahim, Atif Ahmed, Dongrui yang, Fatemeh Saeedi Far, Sher Khan, Yuna Wong, Zongying Liu whom their cooperation whether it's direct or indirect has helped me in my studies.

Finally, My family is the primary influence on my accomplishments. I would like to show my appreciation and gratitude to them. My Father, my mother, my brothers, my sisters, and my wife, Thank you for all the support you have given me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

ABACOC	:	Adaptive BAll COver for Classification.
BoT	:	Bag of Textons.
CNN	:	Convolutional Neural Network.
DenseNet	:	Densely Connected Convolutional Networks.
MR	:	Maximum Response Filter Bank.
STP	:	Semantic Texton Forest.
SVM	:	Support Vector Machines.

CHAPTER 1: INTRODUCTION

In this chapter, the motivation for this work and the objectives are presented and discussed. This chapter presents the organization of this study.

1.1 Overview

Food image recognition has recently gained momentum in the computer vision domain (Yanai & Kawano, 2015). One reason for such motivation is the availability of cheap computing devices within the hands of the public such as smartphones and Internet devices. People with health issues that are related to wrong food intake are becoming more alert about their diets. Therefore, using cameras that can help them identify and keep track of their daily food consumption is useful and essential.

1.1.1 Health Problems related to Food

A reliable assessment of food calories consumption is needed to measure the effectiveness of weight loss interventions and reducing its health-related issues such as obesity, high blood pressure and heart attack likelihood. Obesity is a health problem due to extra energy intake as it is transformed in the humans body as fat (Puhl & Heuer, 2009). Over-gained energy stored can lead to serious health concerns. The automatic food recognition replaces the old-style dietary assessment in accordance with the self-reporting in the food diary, which is frequently inaccurate. Recent works discussed the possibility of dietary assessment through images acquired from the mobile and wearable cameras (Kong & Tan, 2012). Accurate recognition of daily food intake is vital for regulating the food calories consumption and, accordingly, reducing the likelihood of becoming obese.

1.1.2 Image Recognition

Recent success in image recognition is undoubtedly amazing and huge (Rawat & Wang, 2017). Success in fields like image classification, motion detection, and image segmentation provides opportunities to use computer vision in many real-life applications. However, the status of computer vision and pattern recognition is still far from matching human capabilities (Dodge & Karam, 2017). The image recognition system, generally, includes four main phases. It starts with obtaining images, then image preprocessing, image features extraction, and ends with image recognition. Feature extraction is considered as an essential element of the image recognition system.

1.1.3 Food Recognition

Recently, the subject of food recognition for health-oriented applications has gained growing popularity. Among early studies, one study, which was conducted by several authors, who investigated the spatial relationships among various food ingredients. A Semantic Texton Forest (STP) was implemented to section each image into eight various types of an ingredient. After that, pairwise statistics were used to compute a multi-dimensional histogram, and later classified with an SVM. Based on the study, many studies were conducted to find out the optimal hand-crafted representation for food recognition. The Maximum Response Filter Bank (MR) was utilized in a Bag of Textons (BoT) scheme. This representation, which was combined with color descriptors in the nearest neighbor approach, demonstrated that both clues are relevant for the task. The idea of using multiple features was recently brought to the limit by considering as many features as possible and limiting their importance through an ensemble fusion scheme.

Food recognition is important to analyze peoples eating habits through an estimation of calories included in food (Yanai & Kawano, 2015). It is considered a more difficult classification than other image classifications. (Yanai & Kawano, 2015) in their study

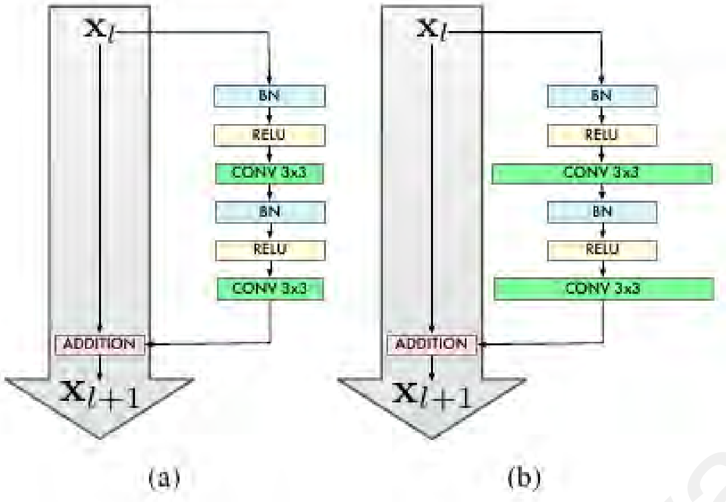


Figure 1.1: Graphical representation of (a) Basic Residual Blocks and (b) Wide Residual Blocks. By expanding the number of convolution kernels (i.e., widening), the number of parameters to learn increases, hence the networks have more capacity.

used a wide residual deep network architecture to solve the food recognition problem.

Figure 1.1 explains (Yanai & Kawano, 2015) architecture with a wide-sliced network. They utilized the batch normalization (BN) and ReLU (ReLU) layers as pre-activations for the convolutional layers (Conv) based on an endorsement from (He, Zhang, Ren, & Sun, 2016) for mapping deep residual networks. The feature maps number in their architecture was expanded as well as the idea of (Zagoruyko & Komodakis, 2016) to increase the representational power of a residual block. Their work showed how they improved the performance of residual networks in comparison with their depth increasing, which addressed the problem with feature dimensioning (Srivastava, Greff, & Schmidhuber, 2015).

1.1.4 Gaps in Food Recognition

Many successes have been recently achieved in food recognition by (C. Liu et al., 2018). However, we are still far from making those works accessible for the daily consumer. First, the conducted studies were based on achieving the state-of-art accuracy for a limited number of food datasets such as FOOD101 (Kawano & Yanai, 2014) and UCEFOOD256

(Kawano & Yanai, 2014). This means that no matter how many food images in those datasets, it will still be limited. Food recognition tends to be more problematic than other categories of image recognition due to the nature of food. A picture of a meal has all different varieties, the difference can be the direction the image that was shot like top or sideways. It can also be the state of the meal, whereas half of it is eaten or only a little left on the dish. These obstacles can represent a huge challenge for conventional image recognition systems. Another difficulty is having a wide range of food categories for new territory the classifier has not explored yet. For example, local dishes for a new country, which are likely to be different than other countries. Therefore, food recognition should be able to support incremental learning with new unseen classes. Recent research has not investigated the possibilities of this field, which encouraged the researcher to investigate these aspects of food recognition.

1.2 Problem Statement

One goal of image recognition and food classification is the ability to learn incrementally without the need to learn or retrain. Despite current research efforts, which provided higher accuracy in identifying food classes, they failed with categories that have a huge number of classes. In food classification, the cost of retraining is high with an increase of food classes every day. This results in increasing the training time, as well as the inability to handle the new types (classes) of food. Accordingly, the problem statement of this study has been substantiated and the main challenges that are addressed include:

- Increasing the training time with huge numbers of food classes.
- Inability to maintain classification accuracy with new food classes.

Many studies were concerned with using the state-of-art methods and approaches for food recognition. Others were concerned with optimizing the performance of a classifier.

However, no study, to the best researchers knowledge, has so far focused on optimizing an incremental classifier for food recognition. The main objective of this study is to apply an incremental classification for food recognition. This studys main aim and objectives are as follows:

1.3 Research Aim and Objectives

Many studies are concerned with using the state of the art methods and approaches for food recognition. Others might be concerned with optimizing the performance of a classifier. However, none are focused on optimizing an incremental classifier for food recognition. The main objective of this thesis is to apply incremental classification for food recognition. In particular, the research aim and objectives are defined as follows:

1.3.1 Aim

- Reduce training time.
- Add new food classes without retraining the model.

1.3.2 Objectives

- To determine which data augmentation methods are best for fine tuning food features.
- To extract features using DenseNet Model.
- To integrate generated DenseNet model food features with an incremental classifier for class-incremental food classification.

1.4 Scope of the Study

In this study, an experiment was carried out with food dataset for food recognition benchmark. The following food datasets were used: FOOD101, UECFOOD-100, UECFOOD-256, and PDFI10. FOOD101 contains 101000 food images for 100 class of western food, which is used to train the deep learning models. UECFOOD-100 and UECFOOD-256 have 100 and 256 classes, respectively of Japanese, whereas PDFI10 has only 10 classes. These datasets were used because of the variety of food categories and most of the studies that have been recently conducted have a benchmark accuracy for them.

1.5 Research Questions

- Does the implementation of different image augmentation methods such as (shear, random crop) produce better features?
- Can an incremental classifier with deep learning features be used to reduce training time for new classes?

1.6 Research Methodology

The problem is addressed by using features from deep CNN. This approach allows for obtaining highly accurate features from Deep CNN, which can result in better classification. Deep CNN is known for state-of-the-art resutls in terms of food classification. Using the last layer of such networks as an image feature is widely used by other works. However, this study proposed to integrate those features with an incremental classifier, which, in return, obtains higher accuracy better than providing a full image to the incremental classifier. Image features can be as small as 300 vector length. However, a full image would be useful if only used in a better resolution, which results in a bigger size.

1.7 Main Contribution

The main contribution of the study is the application of an incremental classifier for food recognition. A new approach is introduced in the study, which is incremental learning that helps in using food data to make food recognition more intelligent. Moreover, the development of hyper-fast incremental classifier model is considered as the main contribution of this research. This model enables the use of different classifiers to integrate with the same features from Deep CNN. Higher accuracy can be obtained from improving incremental classifiers without the need to retrain the Deep CNN model.

1.8 Organization of the Dissertation

This study is organized into six chapters. The first chapter introduces the topic of the study. The second chapter reviews the related literature and previous studies. The models architecture is presented in chapter 3 and the experiments are presented and discussed in chapter 4. Chapter 5 presents and discusses the findings of the study regarding the current benchmark datasets. The conclusion and future work are provided in chapter 6.

- Chapter 1 presents the related domain to this work and an overview of the previous works. It also discusses the problem statement, objectives, questions, and the contribution of this research.
- Chapter 2 discusses images and food recognition in recent works. It shows the current research gap in this domain.
- Chapter 3 consists of the hyperd fast-incremental classifier model architecture.
- Chapter 4 explains the setup of the experiments and the preliminary results of implementing the proposed model.
- Chapter 5 presents all the results regarding the current benchmark datasets.
- Chapter 6 provides the conclusion of the study and discusses the results obtained. It provides the research contribution and describes what is essentially required for future work in this field.

CHAPTER 2: LITERATURE REVIEW

2.1 Overview

This chapter critically reviews relevant studies that are related to the proposed approach in this study. The food recognition problem is regarded as a new area of research in computer vision, as well as pattern recognition. Food recognition like any image classification addresses the features of an image in addition to classifying those features. In this chapter, food recognition is discussed in previous studies, as well as the current features extraction. The current features optimization research is investigated and how it is effective to increase classification accuracy and performance. Deep CNN and incremental learning are also discussed in this chapter as this study focuses on building a hyper model that consists of deep CNN features and an incremental classifier.

2.2 Food Recognition

Following the recent success of deep CNN in image recognition, the most recent works on food, recognition started to achieve new state-of-art results. This involves a different type of methods like food calories estimation (Ege & Yanai, 2017) and location-based food recognition for restaurant menu (Bettadapura, Thomaz, Parnami, Abowd, & Essa, 2015). The most recent work by (Ming et al., 2018) investigated the usage of mobile devices and deep CNN to tackle the key risk of diet tracking. (Ming et al., 2018) developed an algorithm based on Deep CNN to make patients manage their dietary and food intake. This leverage of deep CNN returns to the usage of deep CNN features extraction of food images (Yanai & Kawano, 2015).

Other works started to optimize the neural network architecture to achieve higher accuracy results (Ege & Yanai, 2017)(Pan, Pouyanfar, Chen, Qin, & Chen, 2017)(Zagoruyko & Komodakis, 2016). (Zagoruyko & Komodakis, 2016) differ than other work with a specific convolutional layer that handles the structural peculiarities of some food dishes. (Zagoruyko & Komodakis, 2016) were different from other works as they provided a specific convolutional layer that handles the structural peculiarities of some food dishes. They were the first to utilize the residual learning for food recognition and their usage of many feature maps for each convolution layer to approach the diminishing feature reuse issue in deep residual networks.

(C. Liu et al., 2018) investigated the food recognition problem in terms of developing a new novel deep CNN to deliver state-of-art in food recognition. They suggested using CNN Feature ConcateLayer which is connected to 3 ConvLayer with an extra 3x3 MaxPooling Layer as shown in Figure 2.1.

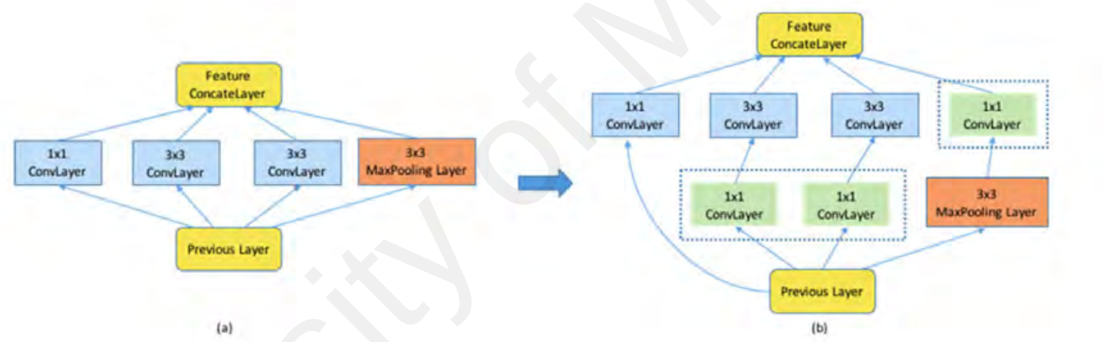


Figure 2.1: Illustration of the Inception Module. Figure (a) in the left is a snapshot of the original network architecture in regular CNN, such as AlexNet. Figure (b) in the right is a snapshot of the new network architecture with Inception Module. The figure is best viewed in color.

They were able to achieve the highest food recognition in terms of accuracy. By comparing their results with the results of other previous studies, the difference is clear as illustrated in Table 2.1. Moreover, (C. Liu et al., 2018) achieved a close to state-of-art low energy consumption.

2.3 Features Extraction

One of the early works on food classification and feature extraction is using local image descriptors (Bossard, Guillaumin, & Van Gool, 2014). This method depends on

Table 2.1: Accuracy table for works in food recognition for UEC-100 with (C. Liu et al., 2018).

Approach	Accuracy
Extended HOG patch-FV	59.6%
D-System (DeepFoodCam(ft))	72.26%
Food recognition system employing edge computing service paradigm (C. Liu et al., 2018)	77.5%

using feature extraction algorithms such as SURF, SIFT, and ORB (Rublee, Rabaud, Konolige, & Bradski, n.d.). Despite the popularity within those algorithms for extracting image features, it did not show any promising results for food classification(Zagoruyko & Komodakis, 2016). Coupling bag of words method with Feature extractor (SURF / ORB) was the common thing in preceding works. which enhanced the accuracy of food images classification. Table 2.3 shows the accuracy obtained by these methods during the evaluation of current research by this study.

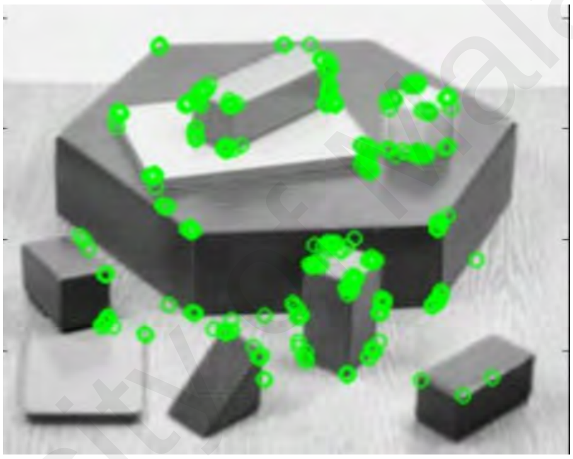


Figure 2.2: Local image descriptors

Method	Dataset	Actuary
BoW	FOOD101	28.51%
SURF BoW-1024	FOOD101	33.47%

Table 2.2: Bow and SURF results with FOOD101

With bounded resource, other studies were designed to operate on mobile phones. (Kong & Tan, 2012) DietCam used a Bag of visual words merged with SIFT to capture features from food images. This model was applied with Support Vector Machines (SVM) classifier. An added factor was the assessment of location for food and extra information about the restaurant yielded better accuracy in (Kong & Tan, 2012).

2.4 Deep Learning Classification

Having big possible images datasets online for different kind of classification obstacles such as ImageNet (Deng et al., 2009)made it plausible to attain higher accuracy in deep learning and deliver success in image classification (Krizhevsky, Sutskever, & Hinton, 2012)(Zeiler & Fergus, 2014). An added element toward the success of deep learning is the employment of the distributed systems and GPU computing on large clusters (Dean et al., 2012). Deep Convolutional Neural Network (Deep CNN) success was examined in competition for image recognition. This competition was for a large-scale dataset, which was ImageNet Large-scale Visual Recognition Challenge (ILSVRC) 2012. Among all the teams that participated in ILSVRC2012,(Krizhevsky et al., 2012) al. were able to win with a comfortable margin to the other teams using Deep CNN. This success was the base for the coming deep CNN networks that encouraged the usage of Deep CNN.Several studies and attempts were made after the success made by (Krizhevsky et al., 2012). (2012) made in ILSVRC. For example, In the subsequent year of the same challenge ILSVRC2013, the winner was (Zeiler & Fergus, 2014). They used a smaller stride of the first convolutional layer and smaller receptive window. These improvements to deep CNN architecture was the foremost focus for following works such as (Sermanet et al., 2013). They worked on improvements in training and testing the model with multiple scales.

2.4.1 Dense Net

DenseNet (Huang, Liu, van der Maaten, & Weinberger, 2016) presented an ability to scale without any downgrades in optimizations. Overfitting or degradation in performance is not an issue with DenseNet due to the consistency of accuracy improvements it produces. It delivered state-of-the-art outcomes across diverse highly competitive datasets in multiple contexts.

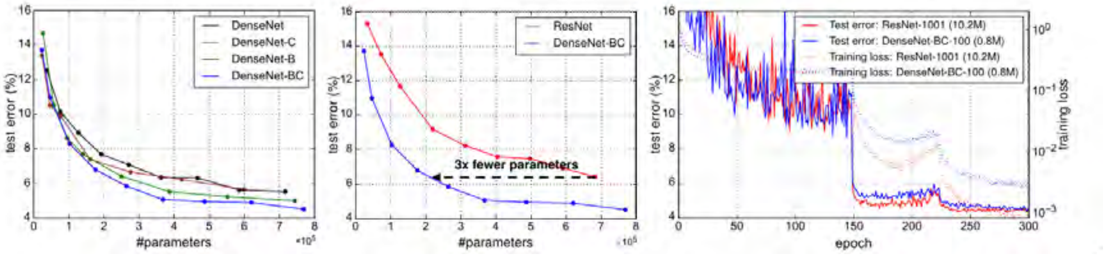


Figure 2.3: Left: Efficiency of parameters Comparison between DenseNet alterations. Middle: Parameters usage comparison with ResNet. Right: Training accuracy comparison with ResNet

Figure 2.3 shows the performance of DenseNet which is equivalent to state-of-the-art ResNets. This comes with Dense requiring notably fewer parameters and computation. DenseNets perform on par with the state-of-the-art ResNets,whilst requiring significantly fewer parameters and computation to achieve comparable performance.

As a primary result of the input concatenation, the feature-maps learned by any of the DenseNet layers can be accessed by all subsequent layers. Dense has compressed internal representations and a reduced feature repetition.This favored feature reuse throughout the network and directs to more condensed models. This means it's valid for features extraction which is the central point of this research.

2.5 Feature selection

Classification networks expects compact and informative data representation (Jeong & Myaeng, 2013). This information is labeled as features and its extraction solves the problem of storing or processing irrelevant data during classification. After extraction, a different operation is conducted to optimize those features and it's termed features selection or features enhancement. Research in this domain, (Blum & Langley, 1997),(Hall, 2000) and (Casimir, Boutleux, Clerc, & Yahoui, 2006), have presented encouraging results in optimizing features without losing any accuracy, on the contrary, it improved the accuracy and reduced the classification time for both training and testing (Borisov, Eruhimov, & Tuv, 2006).

Features optimization has been investigated such as (Jeong & Myaeng, 2013) and (Borisov et al., 2006) and their contribution is considered and explored in this research. The main take of these techniques is the tree-based features selection that showed promising results as mentioned in chapter 3 section 3.4.1.

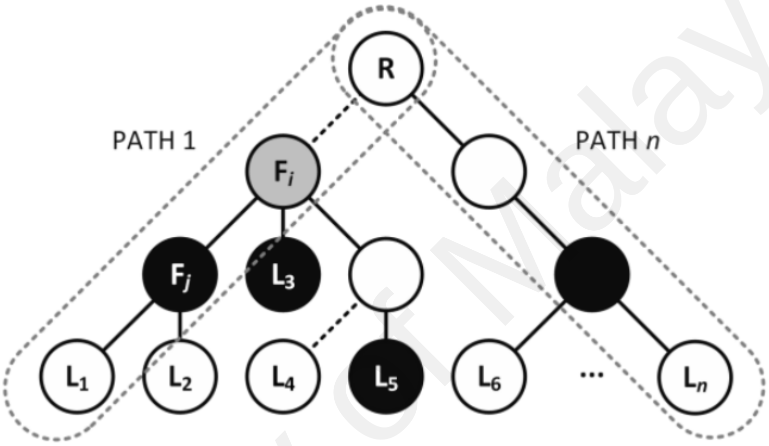


Figure 2.4: Paths connecting the root and the leaf nodes in a feature tree

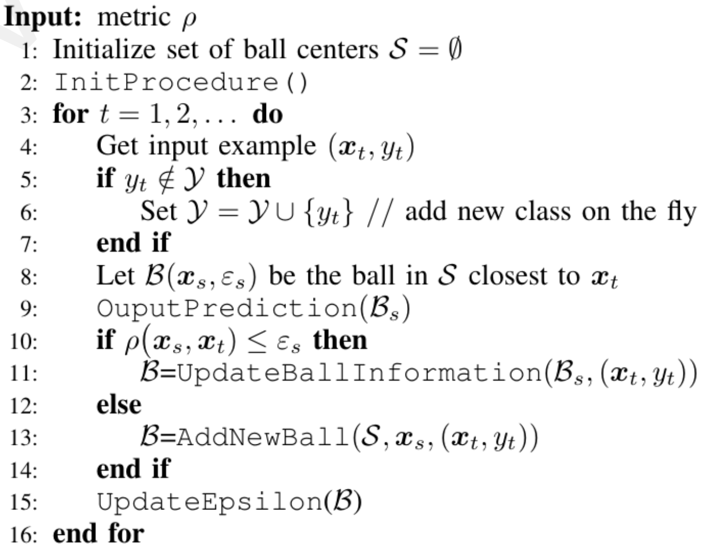
Tree-Based Feature Selection (Jeong & Myaeng, 2013) is an importance based sampling scheme where only a small sample of variables is selected at every step of ensemble construction. The essential approach of the selection algorithm is to assess each of the paths in the tree and pick the relevant feature. The list of nodes between the root and a leaf node is represented as a path. In principle, the problem of selecting features from a tree is converted into smaller problems of selecting a node from individual paths. The process is shown with Figure 2.4 where each node of the tree except the root represents a feature. The tree has n paths resembling the number of leaf nodes. The algorithm selects the most representative node on a path, which is marked with a black node in Figure 2.4.

2.6 Incremental Learning

Recent research in Deep CNN has recorded extraordinary success in various applications in computer vision. Computational complexity is viewed as a significant challenge for Deep CNN. This can rise considerably when attempting to retrain a Deep CNN model. Deep CNN model training requires a comprehensive amount of computational resources (Sarwar, Ankit, & Roy, 2017). Incremental learning replaced it to fix these shortcomings of Deep CNN. It is concerned with the ability to learn incrementally with new streams of knowledge. Its final objective to preserve the previous knowledge that has been learned.However, it can be learned with new examples.(De Rosa, Orabona, & Cesa-Bianchi, 2015) they were able to apply a data streams nonparametric classification using ABACOC algorithm.

2.6.1 Adaptive BAll COver for Classification (ABACOC)

ABACOC (De Rosa et al., 2015) classifies data streams with a novel incremental approach that is nonparametric. They abstract the central concepts behind their approach in a generic algorithmic template (Figure:2.5) called ABACOC (Adaptive BAll COver for Classification), Where x_t is the training data and y_t is the labeled class at given time t.



```
Input: metric $\rho$
Initialize set of ball centers $\mathcal{S} = \emptyset$
InitProcedure()
for $t = 1, 2, \dots$ do
    Get input example $(\boldsymbol{x}_t, y_t)$
    if $y_t \notin \mathcal{Y}$ then
        Set $\mathcal{Y} = \mathcal{Y} \cup \{y_t\}$ // add new class on the fly
    end if
    Let $\mathcal{B}(\boldsymbol{x}_s, \varepsilon_s)$ be the ball in $\mathcal{S}$ closest to $\boldsymbol{x}_t$
    OuputPrediction($\mathcal{B}_s$)
    if $\rho(\boldsymbol{x}_s, \boldsymbol{x}_t) \leq \varepsilon_s$ then
        $\mathcal{B}$=UpdateBallInformation($\mathcal{B}_s, (\boldsymbol{x}_t, y_t)$)
    else
        $\mathcal{B}$=AddNewBall($\mathcal{S}, \boldsymbol{x}_s, (\boldsymbol{x}_t, y_t)$)
    end if
    UpdateEpsilon($\mathcal{B}$)
end for
```

Figure 2.5: ABACOC algorithms template

To elaborate about ABACOC algorithms, Figure 2.5 is a reference template for ABACOC. Although it doesn't specify how each internal algorithm works, however, it displays the common flow for the different types of ABACOC algorithms. Each algorithm will begin with an input metric *p* that is distinct for each type of algorithm. It will constantly start with θ sets of ball centers, S is the set of balls. Next step is running the initial producer, where it can set the initial environment for the algorithm. Y is defiened as the currently known classes. ϵ_s is updated from the *updateEpsilon* producer and its initial value comes from *InitProcedure*, where d_i is the estimated metric dimension. *OutputPrediction* is the producer where it calculates the prediction on each of the algorithms and updating y_t.

Adaptivity to data can be characterized to a specific range with four different methods of ABACOC (called BASE, BASE-ADJ, AUTO, and AUTO-ADJ). Figure 2.6 shows how ϵ_t varies over time, a ball around a center x_s can eventually contain both points assigned to x_s and points non-assigned to it, and even contain other centers.

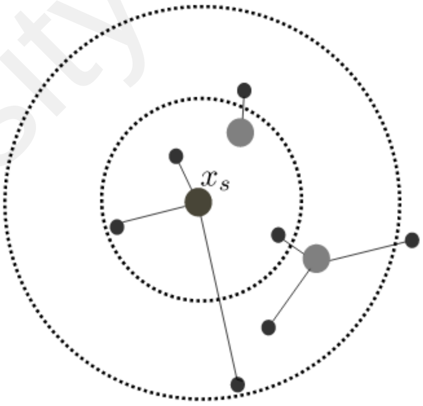


Figure 2.6: Ball representation with Radii

Even though (De Rosa et al., 2015) algorithms are instance-based like nearest neighbour, the learned models are significantly less in size than competing baselines models. When online performance is measured against it, it was found to be more accurate. ABACOC methods (except BASE), are natively multiclass and can support new classes dynamically as they appear in the data stream.

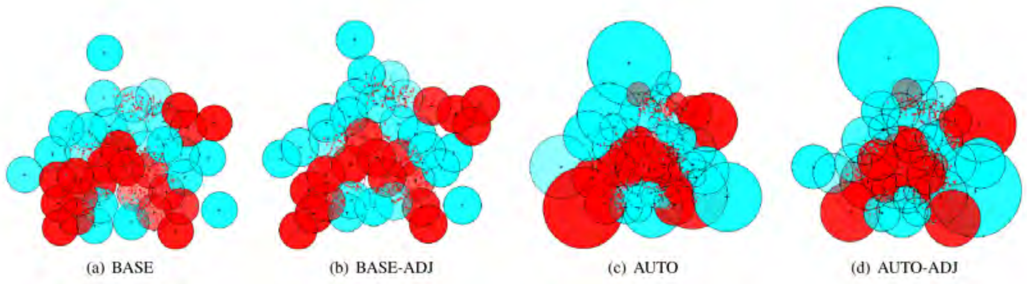


Figure 2.7: Handling data for ABACOC algorthims

For new examples of the data stream that are not covered, the ball goes to the center. They are then classified according to nearest neighbour over the ball center, then each ball predicts many of the labels of previous examples. Figure 2.7 illustrate how observed behaviors of all variants of ABACOC algorithm on 2000 data points of the banana dataset. The magnitude of the color of each ball is proportional to the conditional class probability of the two classes.

According to nearest neighbour over the ball center, new examples are classified, then each ball predicts the majority of the labels of the labels of previous examples. The organization of balls is a tree structure because that will make the computation time logarithmic for each ball. Balls radii shrink to fit new data. Radius shrinking may depend on time or different types of ABACOC algorithms. Decision trees have similarities with ABACOC balls, where leaves are split based on their impurity, ABACOC adapts to the complexity of the model by increasing balls number in input space, where the stream prediction is harder. ABACOC made more improvements handling the relocation of ball centers in the input space due to the nature of their model which is completely incremental. Therefore, balls re-positioning, which made the model use more balls than needed. ABACOC found a solution for this using the K-means step to move the center of a ball being updated towards the median of the data points, which resulted in avoiding a costly global optimization for re-positioning the balls. To sum up, using the input space with balls of possibly different radii ABACOC (De Rosa et al., 2015) can incrementally classify new classes.

2.7 Summary

In this chapter, the food recognition challenges and achievements were presented, as well as recent works on that domain were reviewed. Food recognition has gained more attention with the arrival of Deep CNN success in the computer vision field. The employment of deep CNN made food recognition more plausible for a daily application like dairy assessments and health monitoring for food consumption. The importance of feature extraction in deep CNN was discussed. The first work for feature extraction such as ORB and SURF was reviewed. Although these features are considered enough for certain classification problems, they cannot tackle the food classification problem. This chapter discusses how extracting features from deep CNN model benefited and increased accuracy. DenseNet has been explained in this chapter due to its performance leverage among other deep CNN networks. Incremental learning was also reviewed and how ABACOC (De Rosa et al., 2015) solves problems regarding the performance issue of adapting incrementally for new classes. In summary, Dense net is selected as a feature extractor, which will be trained based on 3 different food datasets. Moreover, ABACOC (De Rosa et al., 2015) success in tackling the incremental issues encouraged the researcher of this study to develop this study classification model to slove catastrophic forgetting.

CHAPTER 3: METHODOLOGY

3.1 Introduction

The model's architecture is presented and discussed in this chapter. The chapter starts with an overall representation of the study's model that explains how the model combines its various components. First, the model's architecture overview describes food images as an input stream for the features extractor. After extracting features, an enhancement is made to reduce the features size. The last step is the food classification. The chapter explains in detail how the proposed features extractor is built. The deep CNN that is used to extract features DenseNet is presented and discussed and how this network is enhanced to obtain more accurate features using image augmentations and multiple food images. DenseNet is a deep CNN with a novel architecture (Huang et al., 2016). It uses the features' maps and blocks between each layer in the network. The use of features, as well as the enhancement process using tree selection, is explained in the first section of this chapter (*1.13. Feature selection scikit-learn 0.20.0 documentation*, n.d.). The next section is phase 2 in the proposed model that is the incremental classification with features extracted from phase 1. This chapter presents the methodology of integrating this incremental classifier with those features and how ABACOC works incrementally with multiple classes in an online stream of data.

3.2 Architecture of the Model

The overall architecture of the proposed model is schematically illustrated in Figure 3.1. The goal of this architecture is to extract features from food images, then performing features enhancement to increase accuracy and performance, and finally using a fast-incremental classifier to classify each food class. This architecture can be modular in each component whether feature extraction or classification. It gives us the ability to integrate new classifiers or build another feature extractor.

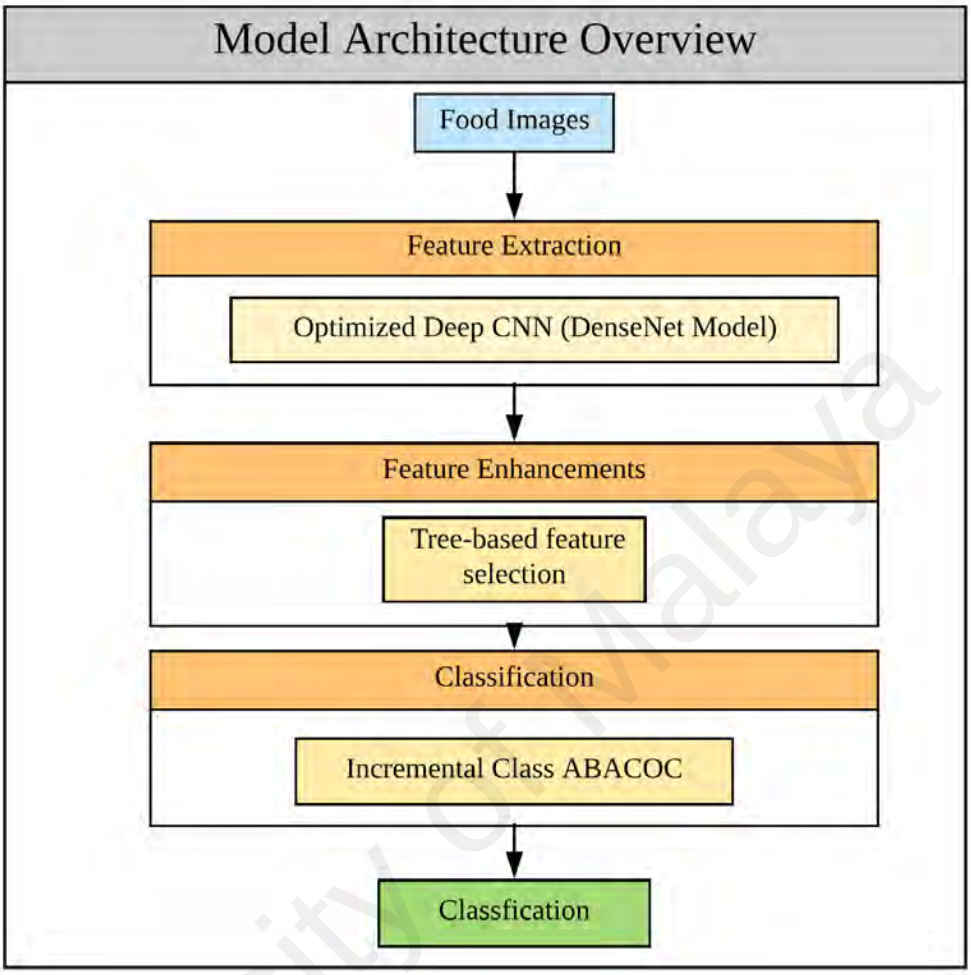


Figure 3.1: Hyper Model Architecture overview.

The hybrid model consists of two general phases. The first phase is building a general feature extraction using Deep CNN, then it is used later for features extraction with new food datasets without the need to retrain Deep CNN. The second phase input is the extracted Deep CNN model features, which do not need to be tightly coupled with classification. This input is food image features that will have to go through enhancement using tree-based (H. Liu & Motoda, 2007) feature selection algorithm. Finally, those enhanced features are ready for classification with Fast Incremental classifier.

3.3 Features Extractor

As illustrated in Figure 3.2, building the food image features extractor consists of 3 components. First, it collects food images from multiple and different food datasets for training. Food images are imported from FOOD101, UECFOOD-100, and UECFOOD-256. Those Datasets are considered as a benchmark for the food recognition problem. The next step involves applying image augmentation to extend the model learning with different food images. The last step involves training the DenseNet Model images taking advantage of DenseNet architecture, which can achieve high accuracy to maximize the feature extraction ability.

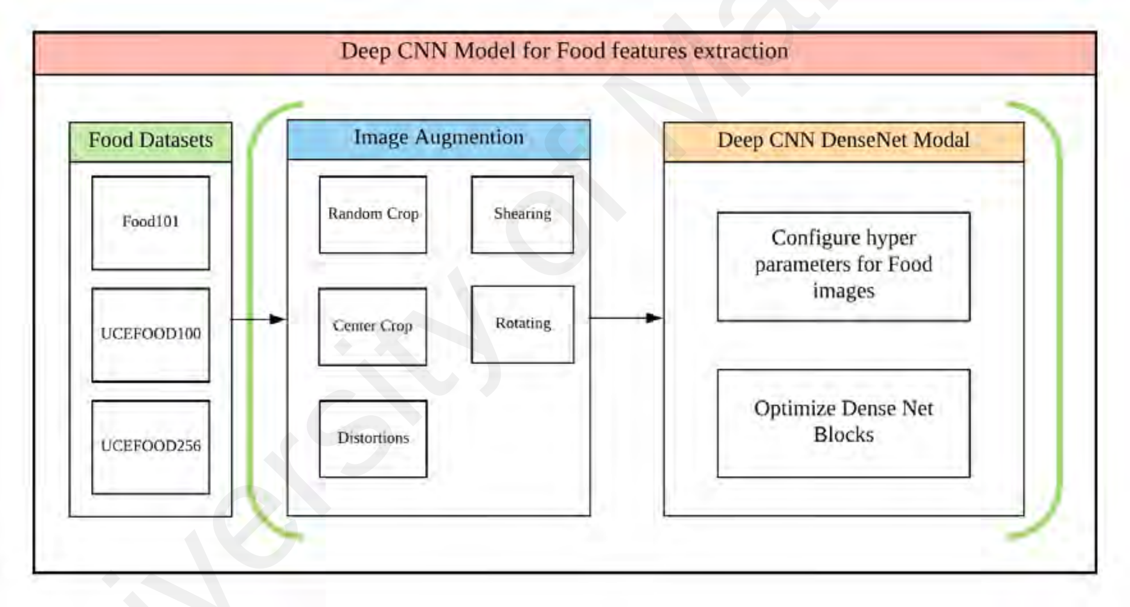


Figure 3.2: Phase 1 Feature Extractor Structure.

3.3.1 Image Augmentation

Image augmentation produces more images of the same image by applying different operations to a single image. The augmented images are generated from a single image using image transformations. For example, you can obtain 3 new images by just rotating the source image by 90°,180°,270°. Another example is distortion or cropping. The focus here on performing operations that are successful with food classification. Cropping,

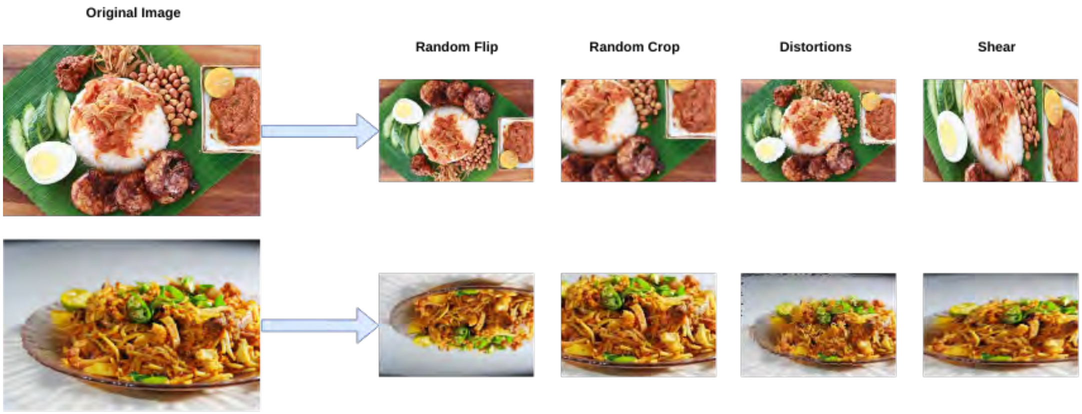


Figure 3.3: Random Image augmentation performed on 2 food images.

rotating, shearing and random distortions were used. It enables a single food image to have multiple instances that can be a result of taking pictures of the same food in the wild. For example, a group of 4 people taking pictures of the same dish can lead to having 4 rotations of the exact same food. Augmenting food images using those methods lead to a more accurate model, which leads to better food features. Further discussion is provided on chapter 4 section 4.4.

3.3.2 Deep CNN - DenseNet

DenseNet has reliable design and architecture, which allows obtaining the state-of-art with fewer resources. It consists of layers as blocks instead of the traditional deep CNN layers. This means it has fewer layers. It uses the same feature map size for each block with pooling to reduce it before each block input shown in Figure 3.4. It encourages features reusability; feature propagation is strengthened and hugely reduces the parameters' number.

DenseNet model was trained with 3 dataset (food101 , UECFOOD100, UECFOOD256). Dataset preparation is further discussed in chapter 4 section 4.2.1. Once our Deep CNN model reaches a high-test accuracy we save weights of that model to later trim the last pooling layer from the DenseNet Model. Therefore, we can have the features' layer that

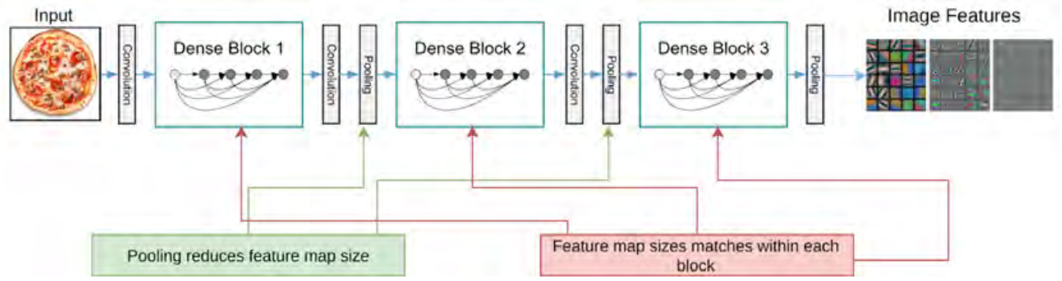


Figure 3.4: DenseNet model architecture overview.

will be used as an input for the next phase. The last pooling layer will be used as a feature vector for food images.

3.4 Incremental Classification

Combined feature extraction with an incremental classifier is shown in Figure 3.5. The first is the extraction of features from pre-trained model classes will go to feature extraction. Next is performing features enhancements (H. Liu & Motoda, 2007), which limit the number of features used in the classification stage. Based on the features incremental classification will train on new features for new food classes without the need to retrain.

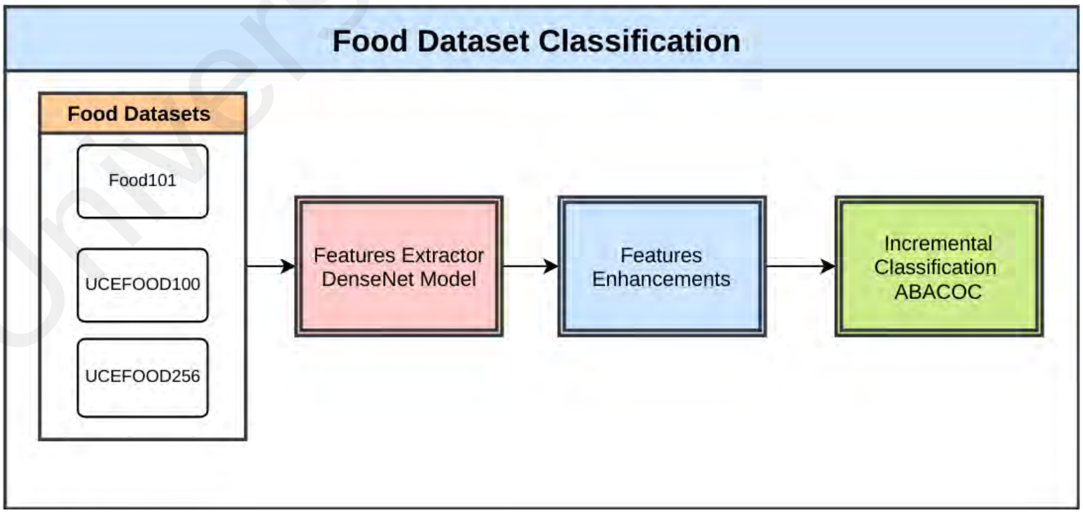

Food Dataset Classification
Food Datasets
Food101
UCEFOOD100
UCEFOOD256
Features Extractor DenseNet Model
Features Enhancements
Incremental Classification ABACOC

Figure 3.5: Phase 2 Food classification.

3.4.1 Feature Enhancements (Tree-Based Feature Selection)

Tree-based estimators (H. Liu & Motoda, 2007) can be used to compute feature importance, which in turn can be used to discard irrelevant features. This experiment shows the use of forests of trees to evaluate the importance of features on an artificial classification task. The red bars are the feature importance of the forest, along with their inter-trees' variability.

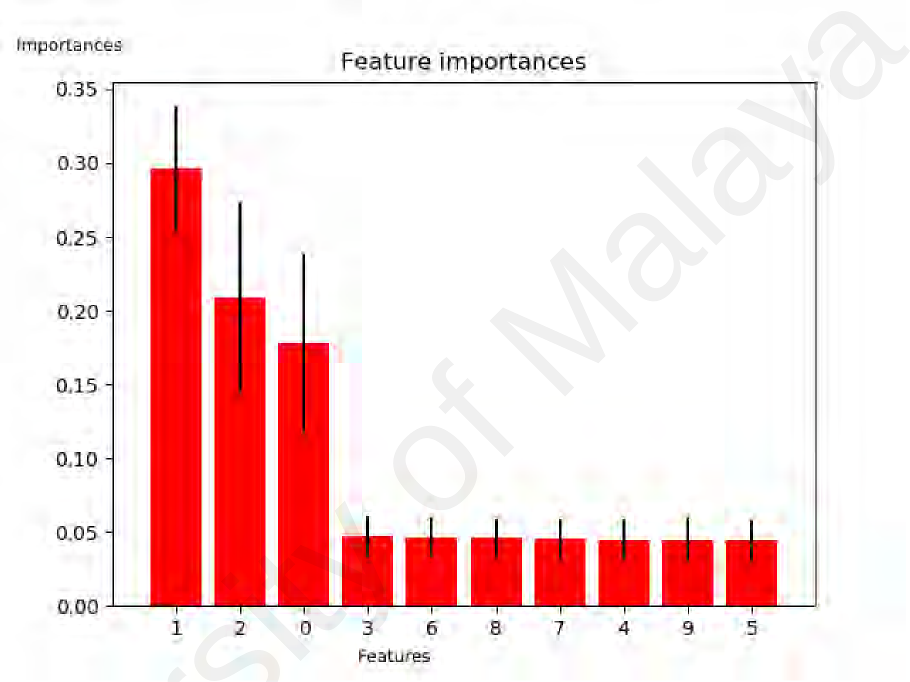


Figure 3.6: Features importance.

As expected, Figure 3.6 suggests that 3 features are informative, while the remaining are not viable. The utilization of tree-based feature selection can reduce the size of our feature vector from 1664 elements to 1080. This allows us to make the classification more efficient regarding performance and speed.

3.5 ABACOC Algorithm (Incremental Learning)

ABACOC is a model that covers the input space utilizing simple local classifiers. The distribution of the classifiers has dynamically adapted to the local (unknown) complexity of the classification problem. ABACOC struck a good balance between the model complexity

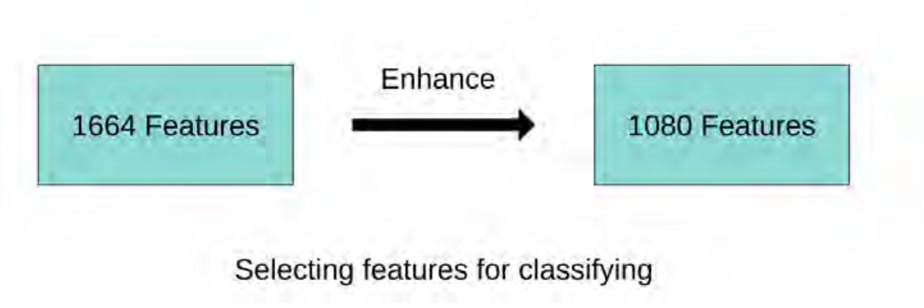


Figure 3.7: Example of features enhancements

and the predictive accuracy. Figure 6 shows how ABACOC performs when new classes are added to the model. The model accuracy does not clearly decrease dramatically. However, this decrease can be addressed in future work.

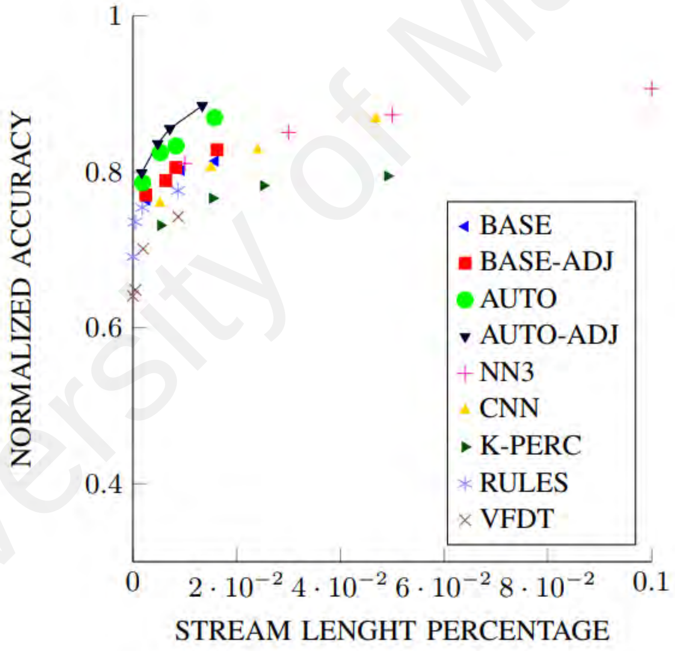


Figure 3.8: ABACOC Incremental accuracy comparison

Images samples number can play a major role in improving ABACOC model accuracy. This improvement can lead to better accuracy. However, the model accuracy starts to stabilize with time. Image samples cannot have that effect on the model performance after that stage, which becomes problematic. Alternative incremental classifiers can provide a solution to this problem since food features are independent of the classifier.

3.6 Summary

To summarize, This chapter included a concise discussion about the methodology and the proposed model of this study. It starts by addressing the overall precipitation of the final proposed model. It then goes deeper with an explanation of each component of the model. The first component is the feature extractor. It contains 3 main sections, the selection of the food datasets, Augmenting images and Training Deep CNN model to extract the features. Once we combined those sections we get a feature extractor that has the capacity to generate image food features. The second Component is utilizing the features extracted from the first component in an incremental system. We begin this process with features enhancement before classification. Incremental learning is briefly explained with the desired algorithm (ABACOC).

More details on the decisions that lead to this proposed model are discussed in the next chapters.

CHAPTER 4: EXPERIMENTS

4.1 Introduction

This chapter presents and discusses the research experiments and the tools that are used for conducting these experiments. It explains the setup for the research environment and food datasets. It also presents the structure of the datasets and the characteristics of each one of them. The preliminary experiments are presented using standard methods for image classification. The usage of Deep CNN features is specified for images starting with extracting features of current pre-trained Deep CNN model weights. The chapter explains in detail the testing with fine-tuning a Deep CNN model on food images. The methods that are used for food image augmentation, which help increase the model overall accuracy are introduced. The final section of this chapter describes the process of integrating an Incremental Classifier with food images. Integrating this classifier includes enhancing image features using different features of selection algorithms. It also includes using different incremental classifiers. The flow of the proposed model is presented and discussed starting with the first phase until the classification part.

4.2 Scope of Experiments

The scope of experiments is limited to food images. Recent studies provided a benchmark food dataset for training and testing (Akhi, Akter, Khatun, & Uddin, 2018). These datasets are used in this study as a guideline for the conducted experiments. The experiments were conducted using a PC desktop server with GPU support. The study server GPU is NVidia Tegra K1 with 16 GB shared memory and enabled CUDA drivers for parallel computing. Using CUDA allowed the researcher to perform feature extraction much faster than CPU (Zhang, Fang, Zhou, Pan, & Cong, 2016). The experiments environments are on Linux using Python and MATLAB as the programming languages in this study.

4.2.1 Food Datasets

Food images were explored as recent works were concerned with 3 main food datasets. Food101 (Bossard et al., 2014), UECFOOD-100(Matsuda, Hoashi, & Yanai, 2012) and UECFOOD-256 (Kawano & Yanai, 2013). These datasets provided enough images that are required for Deep CNN model training. They provided a variety of food classes from different regions. Images were partitioned to 2 main groups: Training and Testing. This partitioning was made based on the same procedure provided from the authors of these datasets.

4.2.1.1 FOOD101

This dataset introduces challenging data that contains 10,000 images for each class. Hence the name it has 101 different food categories. It provides a manual selection of 2500 images for each category that is defined as test images. The other 7500 are training images. Another challenge is that dataset training images have some noise. The noise is defined by some wrong labeling or coloring. These dataset images are scaled to a maximum width of 512 pixels. Overall, 101,000 images of 101 different food class can be considered a fair amount of data for Deep CNN training (Bossard et al., 2014).

4.2.1.2 UECFOOD-100

UECFOOD-100 is different from FOOD101. One major difference is that it comes with the bounding box of each meal location in the images. This helps in cropping images to make the classifier clearer and, therefore, it removes all the surrounding frames of pictures such as food table. It was collected based on the most popular Japanese food. It comes with 100 unique food classes. It is used with a mobile application for food recognition in Japan. One of the limitations of this dataset is that it comes with an inconsistent number of images for each class. Some classes have more than others, which can lead to overfitting of Deep CNN (Matsuda et al., 2012).

4.2.1.3 UECFOOD-256

This Dataset is considered an extension of UECFOOD-100. It is also part of Japanese food categories. It has all the food class of UECFOOD-100 with an extra 156 category. It challenges Deep CNN classifier due to the increasing number of classes. It has the same issues as UECFOOD-100 with inconsistent images distribution of each category (Kawano & Yanai, 2013).

4.3 Preliminary Experiments

First, the experiment was carried out with the current work and algorithms in image classification and feature extraction field. This allows us to explore the current state-of-the-art (SOA) algorithms of Deep CNN features extraction together with well-known algorithms for instance ORB and Bag of Word (Farinella, Moltisanti, & Battiato, 2014) (Rublee et al., n.d.). The preliminary experiments were made with 200 images of UECFOOD100. We started experimenting with a visual bag of words technique using K-means, then using ORB as linear (Wagsta, Cardie, Rogers, & Schroedl, n.d.). After that, we explored the possibility of using Deep CNN features as an input from a pre-trained model on the image net.

4.3.1 K-means Bag of Words ORB

The first experiments involved K-means and Oriented FAST and Rotated BRIEF (ORB). ORB is used to extract features from food images (Rublee et al., n.d.). ORB is basically a blending of FAST keypoint detector and BRIEF descriptor with many adjustments to improve the performance. It employs FAST to find key points, then implements Harris corner measure to discover top N points among them. It also uses the pyramid to produce multiscale-features. But one obstacle is that FAST doesnt calculate the orientation.

Therefore, what about rotation invariance? Authors obtained the following alteration. K-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. K-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. Later, k-means clusters are used as words for BoW to create a histogram for each food image. Computed Histograms are now available for classification. Next step is fitting those histograms for a classifier to determine classifications accuracy. For fast and accurate classification, high-performance Extreme Learning machine was applied. HP-ELM is a high-performance Neural Network toolbox for solving considerable problems, especially in Big Data. It supports datasets of any size and GPU acceleration, both with modest memory consumption and fast non-iterative optimization; training a neural network with 32000 neurons on MNIST dataset in 1 minute on your desktop. Using these does not provide enough accuracy results. Table 4.1 illustrates shows that the test accuracy for this method is 10.0% and it's considered as low accuracy in computer vision. This happens because the extracted features from Bag of words are similar despite being different pictures. Similarity comes from the nature of ORB which is unsupervised and not tuned for food images.

Images	Training Images	Test Images	Train Accuracy	Test Accuracy
14,360	10,052	4308	22.3%	10.0%

Table 4.1: Results of testing with K-means combined with Bag of words

4.3.2 Deep CNN Model with ImageNet Weights

The convolutional neural network CNN model was used to extract features from before the last layer in the network (Deep CNN Features without fine-tuning). We considered VGG as our first model for feature extraction (Simonyan & Zisserman, 2014). VGG

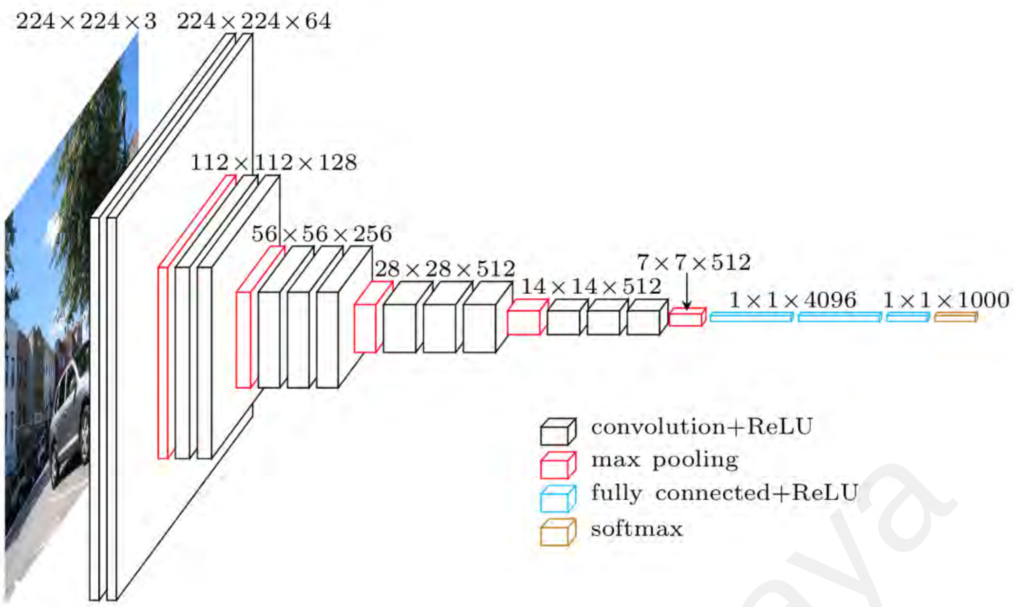


Figure 4.1: VGG architecture

is a convolutional neural network model for image recognition proposed by the Visual Geometry Group in the University of Oxford, where VGG16 refers to a VGG model with 16 weight layers, and VGG19 refers to a VGG model with 19 weight layers. Figure 4.1 illustrates the architecture of VGG16: the input layer takes an image in the size of (224 x 224 x 3), and the output layer is a softmax prediction on 1000 classes. From the input layer to the last max pooling layer (labeled by 7 x 7 x 512) is regarded as the feature extraction part of the model, while the rest of the network is regarded as the classification part of the model.

By using the pre-trained model loaded with ImageNet weights, the features were extracted from both datasets FOOD101 and UECFOOD-100. These features are based on each class or category of food. We used the same classification algorithm that we used with the previous experiments. HP-ELM is our choice for classification due to the reasons that were mentioned above such as speed and performance.

Using this method we were able to have accurate results. Results table below shows accuracy using this method. On FOOD101 we are able to get a result near a state of art without any training to ImageNet model.

By using this method, accurate results were obtained. Table 4.2 illustrates the accuracy results using this method. On FOOD101, we obtained results that are close to the state-of-art without any training to ImageNet model.

To elaborate on these results, It is shown that supervised learning with large data obtains higher accuracy. Taking in hand the availability of food images to make the classifier more intelligent.

Dataset	Classes	Images	Training Images	Test Images	Train Accuracy	Test Accuracy
Food101	101	101000	70700	30030	95.32%	80.96%
UECFOOD-100	100	14360	10052	4308	96.437%	70.79%

Table 4.2: Comparison for two dataset with Deep CNN features

Based on Results from table 4.2 we are able to identify the desired method for our research and to proceed with it.

4.4 Image Augmentation

To make the most of the few training examples, they will be augmented through several random transformations so that the proposed model never sees twice the same picture exactly. This helps in preventing overfitting. It also helps the model in generalizing better. Given the available limited images, transformations, which include rotation, translation, and scaling, were carried out to increase the training data. Upper bounds were set up to perform the transformations, within which each image was haphazardly subject to these transformations. This substantially increased the data size. Figure 4.2 shows examples of raw images and expanded images examples. When it comes to image augmentation, the rule of thumb is to randomize the process as much as possible. This leads to generated food images that can represent the real images captured from a mobile camera. Rules have been set to limit the randomization of image augmentation to make sure that it doesn't produce augmented images that are not possible to be found in real life . for example,

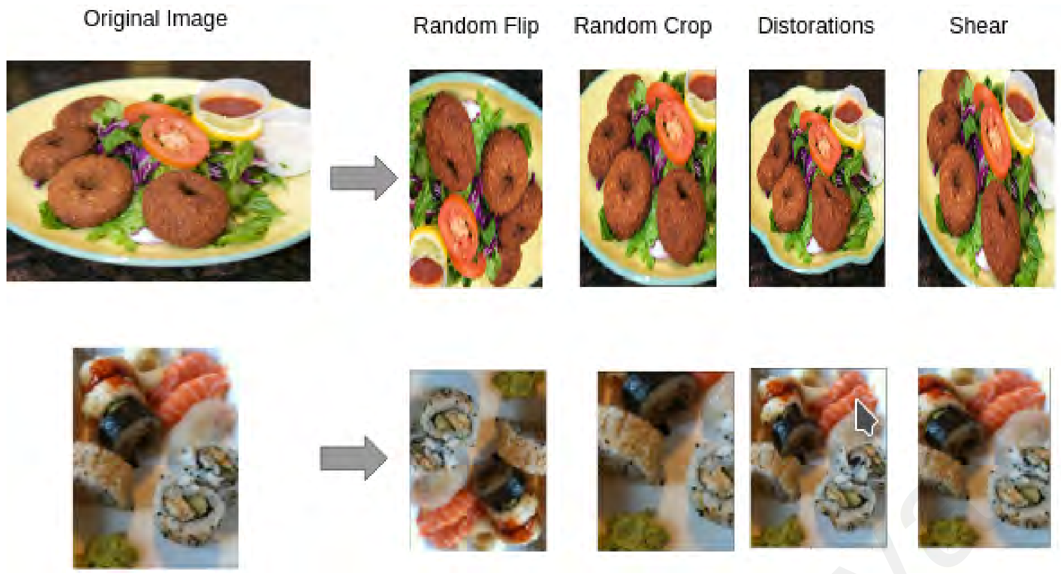


Figure 4.2: Examples of Image Augmentations

having high magnitude of distortion can lead to non-viewable image. Experimenting with 3 classes and with 10 food images, it was noticed that the models accuracy increased when using the following types of image augmentations:

- Elastic Distortions :

  It's been applied with a probability of 100% to assure this augmentation will accrue, grid width (The number of rectangles in the grid's horizontal axis) of 4, grid height of 4 rectangles and with 8 rectangles magnitude.

- Random Rotations

  Random rotations have no limit. It can be within 0 to 360 degree with a probability of 100%.

- Random Cropping

  When cropping it was determined that the percentage area of cropping should be more than 90% of the original image. This boundary was set to contain the possible details of the food image due to the nature of food images. Food images usually are taken with the full dish in the image, therefore, there is no reason to crop less than 90% .

- Size Preserving Shearing

  The probability for this augmentation was set to 100%. It had a limit of 10 degrees to the right and 10 degrees to the left. Limits applied to this augmentation meant it preserved its validity as food image.

Later, the food images were used from the available datasets to make sure that the proposed model can extract and recognize food features.

4.5 Preparing Features Extractor

Multiple food data were combined with similar classes to train the proposed model for feature extraction of that specific class, which allowed us to have more samples for each class. Using DenseNet model for training made it possible to ensure high accuracy; training the Deep CNN model on the combination of both (UECFOOD-256, FOOD101) datasets images for over 87 hours to achieve high accuracy. This selection of UECFOOD-256 and FOOD101 is made to insure that the Deep CNN model is exposed to variety and challenging food images. When training was over, results were found to be satisfying as features could be extracted from before the last layer.

4.6 Features Enhancement

Tree-based feature selection is used to enhance the extracted features from DenseNet model. The experiment was conducted on UECFOOD-100 Dataset with cropping that has 14,358 of food Images. This dataset was split into train and test categories, whereby train images are 10768 and test images 3590.

We classify using High-performance ELM (HP-ELM) for enhanced and non-enhanced features. Based on Table 4.3, better accuracy was achieved on both train and test section. This accuracy produced by tuning the features is due to multiple reasons. First, enhanced features have been trimmed out of unnecessary noise that could affect the classifier. Sec-

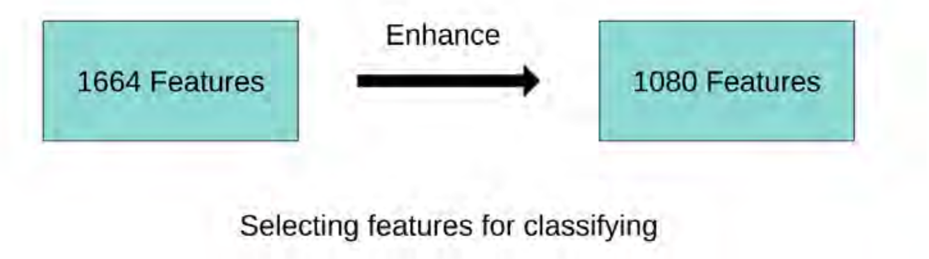


Figure 4.3: Features enhancements

ond, it enhances generalization by reducing overfitting (formally, reduction of variance). Lastly, Some data could have either repetitive or unnecessary features. It can be dismissed without any waste of information, and this is feature selection. Repetitive and unnecessary are two distinct notions, since one relevant feature may be repetitive in the presence of another relevant feature with which it is completely correlated.

Type	Size	Train Accuracy	Test Accuracy	Time
Original	1664	94.2%	73.1%	1.39 seconds
Trimmed	1082	94.4%	73.96%	1.18 seconds

Table 4.3: Comparison of features accuracy with enhancements

Although this results shows only 0.2% increase in training accuracy , 0.86% increase in test accuracy and 0.21 seconds saved of training time, It is important to mention how it's now smaller in size which makes efficient for low memory devices and better with features enhancements when applied on large items. As shown in Table 4.3, it is 34.9% ((new size / original size) - 1) smaller in size and 15.1%((new time / original time) - 1) faster compared to non-enhanced features. These percentages are relative to the number of images (14,358) mentioned above.

4.7 Incremental Learning Algorithms

In terms of classification for the features, we tried several Incremental Learning Algorithms to obtain the highest results. Recent work has given us ABACOC (De Rosa et al.,

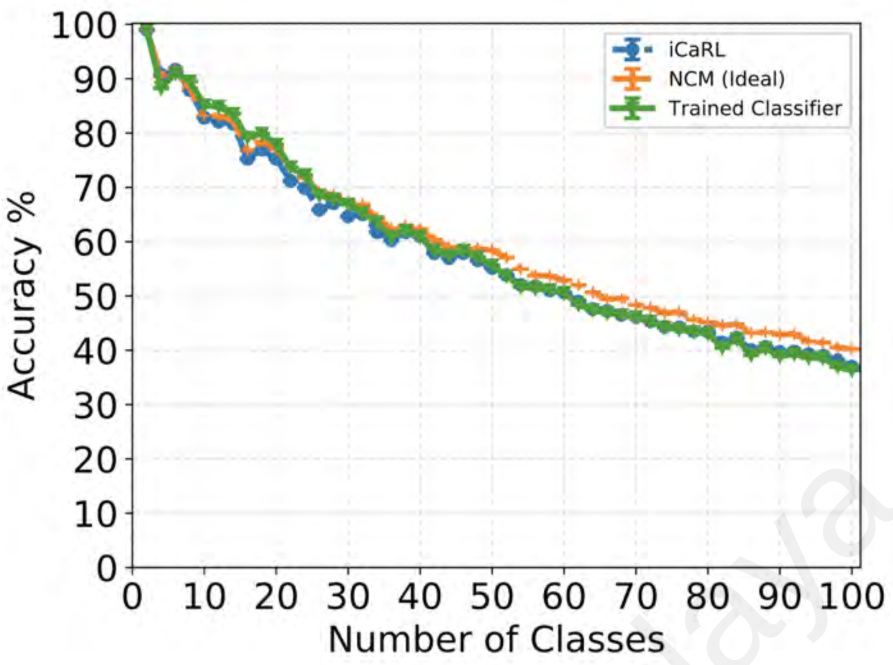


Figure 4.4: Incremental Learning classifiers comparison

2015), Learn++ (Polikar, Upda, Upda, & Honavar, 2001) , Incremental Extreme Learning Machine (IELM)(Convex incremental extreme learning machine - ScienceDirect, n.d.), iCaRl (Rebuffi, Kolesnikov, Sperl, & Lampert, 2017) and Incremental Support Vector Machine (ISVM) (Cheng & Juang, 2011). As the figure 4.4 shows how iCaRl maintain accuracy within the increasing number of classes.

In terms of classification for the features, we tried several Incremental Learning Algorithms to obtain the highest results. Recent studies have provided us with ABACOC (De Rosa et al., 2015), Learn++ (Polikar et al., 2001), Incremental Extreme Learning Machine (IELM) (Convex incremental extreme learning machine - ScienceDirect, n.d.), iCaRl (Rebuffi et al., 2017) and Incremental Support Vector Machine (ISVM) (Cheng & Juang, 2011). As shown in Figure , the way iCaRl not drastically dropping accuracy within the increasing number of classes is illustrated in 4.4.

4.7.1 Incremental Learning Classifier with iCarl

The experiment was conducted with iCaRl on the food dataset features. The same features extracted from DenseNet model were used with different incremental algorithms. A small sample of our dataset features was used testing only on 10 classes from FOOD101 dataset.

Dataset	Classes	Images	Training Images	Test Images	Train Accuracy	Test Accuracy
Food101	10	10100	7070	3003	36.4%	16.96%

Table 4.4: iCarl accuracy results

Table 4.4 illustrates the results of using iCaRl algorithms with the samples gathered. It is not acceptable since it only achieved a 16.96% test accuracy.

4.7.2 An Incremental Classifier with ABACOC

The experiment was repeated with Adaptive Ball Coverage Classification ABACOC Algorithm. The same data sample was used as an experiment with iCaRl.

Dataset	Classes	Images	Training Images	Test Images	Online Accuracy
Food101	10	10100	7070	3003	84.3%

Table 4.5: ABACOC accuracy results

Table 4.5 displays the results of applying ABACOC algorithm including the collected samples. It presents an encouraging result as 84.3% Online Accuracy was achieved (De Rosa et al., 2015).

Figure 4.6 demonstrates how the samples number affects accuracy in online incremental learning. Accuracy started stabilized and slowly increased with increasing the number of samples. That can benefit if a dataset is increasing with time.

4.8 Summary

To review this chapter, It was listed all the experiment conducted in this research and various methods that lead to the proposed model. This chapter begins with an introduction

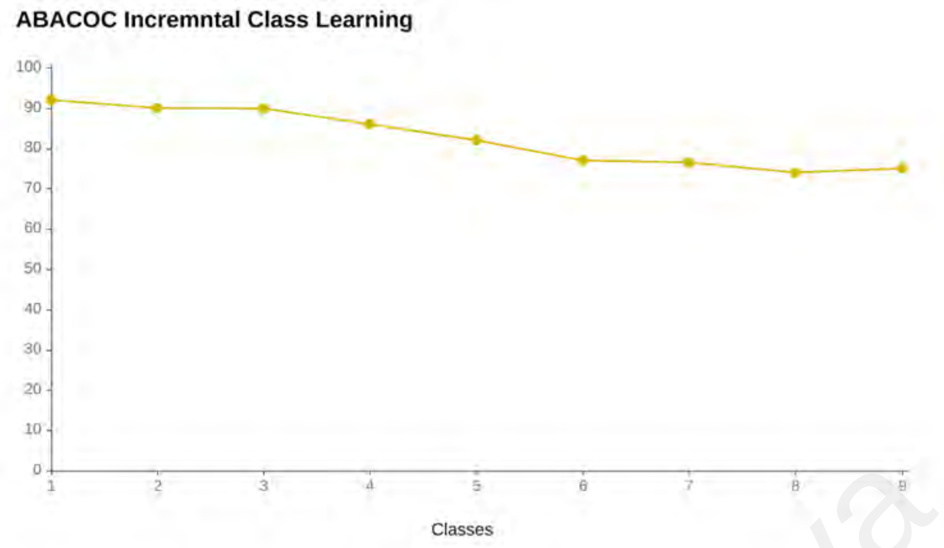


Figure 4.5: Classification accuracy when adding new classes to model

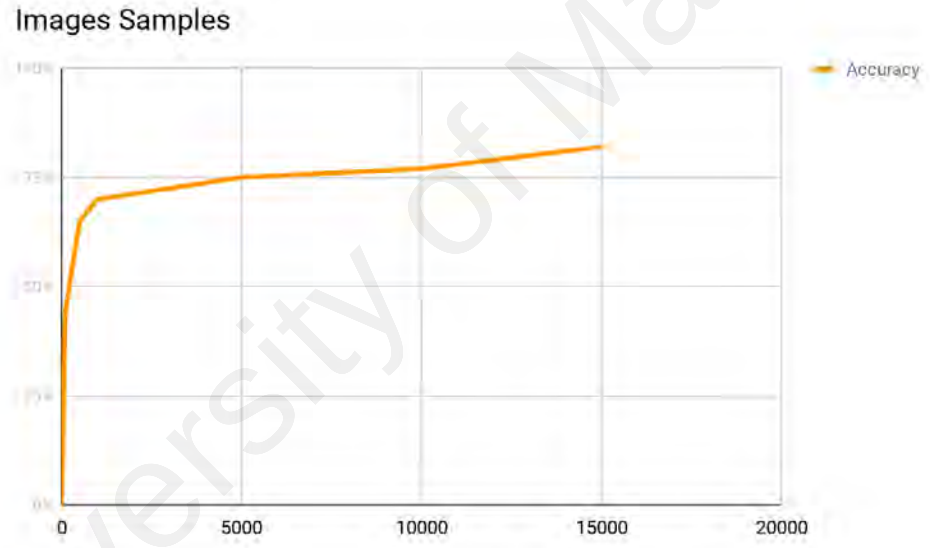


Figure 4.6: Classification accuracy when adding Samples to model

to lay the foundation of how the chapter is arranged. It then renders the scope of which this research has to conduct the experiments.

Inside the scope of experiments, Food datasets are reviewed. It illustrates the three major food datasets for food recognition problem which are FOOD101, UECFOOD100, and UECFOOD256.

Preliminary experiments were detailed and addressed in this chapter. It conveys the reasons why Deep CNN was chosen for features extraction rather than conventional feature extraction such as ORB.

Image augmentation has been investigated. The capacity to generate more image computationally without the lack of gathering real images was the purpose of Image augmentation. It was mentioned what types of augmenting, rules and limits were performed in this research.

The final section of this chapter was the introduction of features enhancement and Incremental Learning. Feature enhancement results have shown an improvement rather using unenhanced features. Incremental learning algorithms were explored in this chapter. experiments were performed on iCarl, ABACOC, and IELM. ABACOC performed better results thus was selected for this study model.

CHAPTER 5: RESULTS AND EVALUATION

5.1 Results

By using the proposed method, satisfying results were achieved. The model was trained with Optimized DenseNet model for food images and categories. The trained model was used to extract features of food images by trimming the last layer of the network, which is the pooling layer of classification. Features will go through enhancements using the tree-based selection of features. This enhancement will make the features size smaller, which affects the classification time and performance. Classification will be done using adaptive Ball Cover Classification. This online algorithm uses online learning with local classifiers to manage accuracy stabilization.

The experiments included Online and Incremental learning by classifying 10 classes of food images. As shown in Figure 5.1, the accuracy is high and acceptable at 84% with a training time of 724.06 seconds.

Besides the current knowledge of 10 classes, the other 10 classes were added to the model without retraining it. For the first samples, the accuracy dropped significantly introducing new classes. However, with more samples of the new 10 classes, the model started to increase the accuracy overall and was stabilized in the end. On the other hand, the training time is 653.23 seconds. This shows that the model does not need more time for new classes even though they are new ones. Figure 5.2 shows the increasing accuracy with increasing the number of samples.

5.2 Evaluation

The results showed the proposed model with incremental learning using different benchmarks datasets. The method of extracting features was used for each dataset using the same trained model without retraining the model for each one. This means only

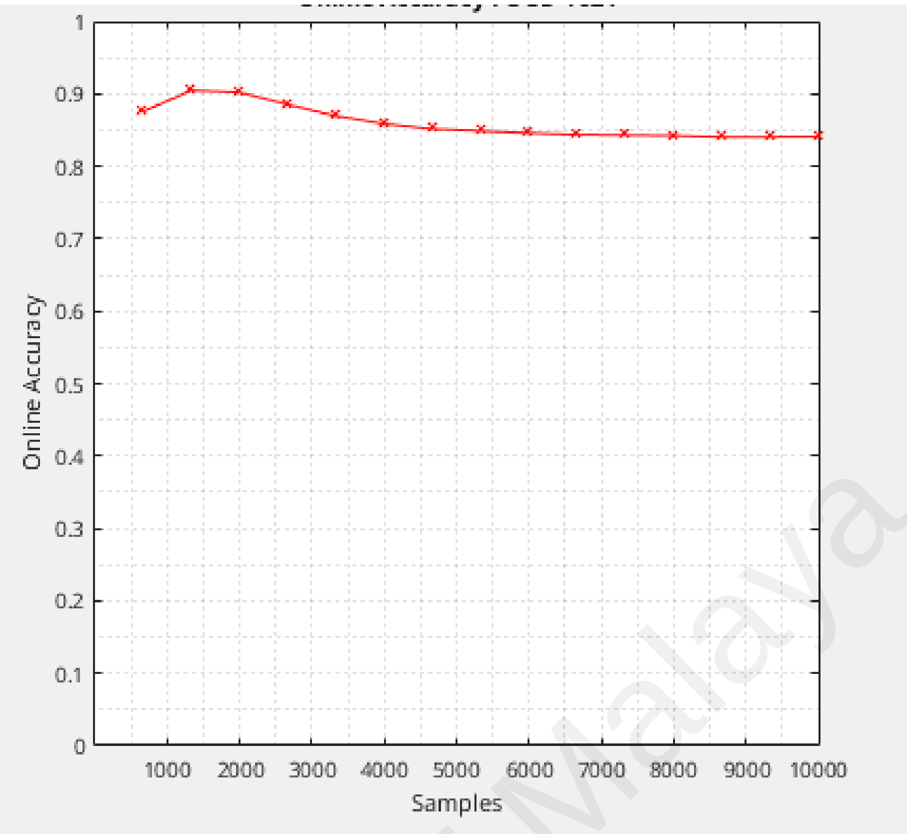


Figure 5.1: Online Classification accuracy for 10 classes

incremental learning classifier was trained on the datasets features. This study results is compared with (C. Liu et al., 2018) work. They have achieved the state-of-art in food recognition. In contrary to this work approach, they have used a traditional deep CNN for classification. It roughly takes **2 to 3 days nonstop** for a server equipped with Nvidia K40 GPU to train the model due to the large dataset of food images and their augmented images. This large dataset means that the classifier have the required knowledge to recognize food classes. However, Deep learning classification require heavy computational power for each iteration.

Dataset	**Top-1**	**Top-5**
FOOD101	77%	94%
UECFOOD-100	76.3%	94.6%
UECFOOD-265	63.6%	87.0%

Table 5.1: (C. Liu et al., 2018) Food datasets accuracy results

Table 5.2 illustrates that training times are in a sufficiently great way lower than what

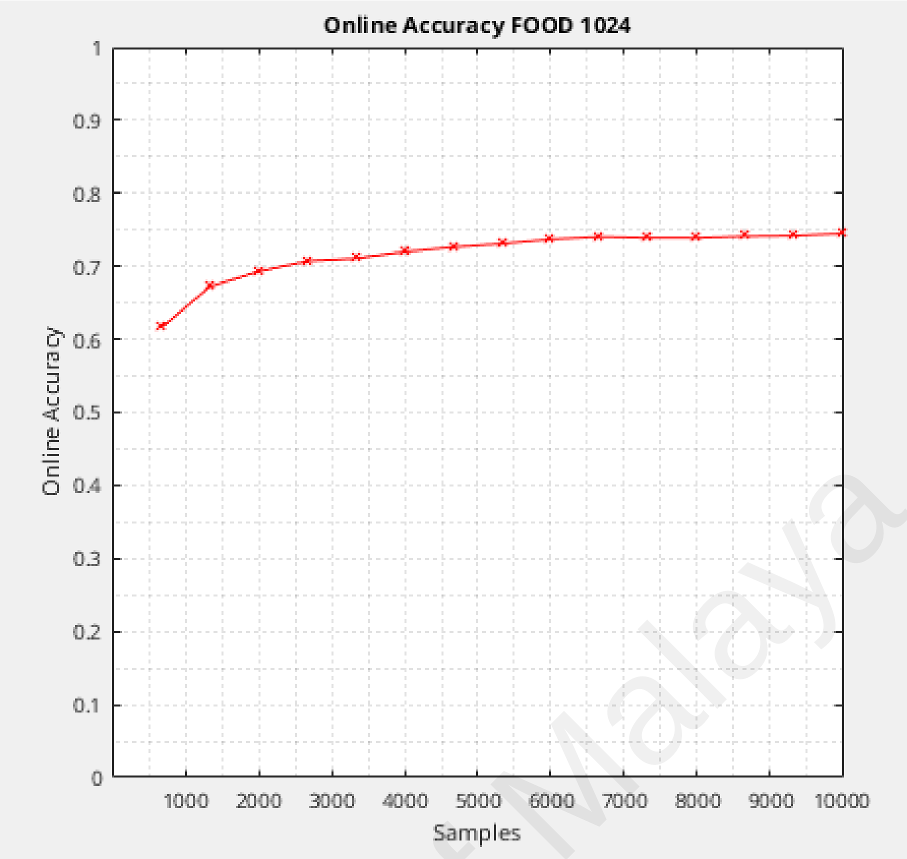


Figure 5.2: Online Classification accuracy for all classes

is in deep learning models. However, accuracy results are not as high as the state-of-art for those datasets. Future work can focus on increasing the accuracy of each dataset benchmark. Moreover, future work can also implement different incremental classifiers to be able to run efficiently on mobile platforms or cloud infrastructure.

Dataset	**Feature Extraction Time**	**Training time**	**Accuracy (Top-1)**
FOOD101	80.23 Seconds.	1253.36 Seconds.	%77.72
UECFOOD-100	40.3 Seconds.	816.72 Seconds.	%63.72
UECFOOD-265	55.3 Seconds.	936.72 Seconds.	%59.7

Table 5.2: Food datasets accuracy with this study approach

CHAPTER 6: CONCLUSION

The food recognition problem is regarded as vital for health problems, which are related to food consumption and dietary assessments. Recent studies have provided enough knowledge in this field. However, they missed the ability to intellectually update an existing model with new data whether they are new food samples or classes.

6.1 Contribution

This study introduced A Hybrid Deep CNN Model for Fast Class-Incremental Food Classification. Using this model will add new classes to the proposed model by extracting features of those respective class images. The features of FOOD101 dataset were extracted and classified in approximately 20 minutes using the proposed method in this study with 77% accuracy. This resulted in **faster training time** for new classes or datasets regardless of the number of classes or images due to unnecessary training time of the studys deep model. It is presented when using the same model for UECFOOD-100 dataset, which the proposed model has never seen. It does both features extraction and classification in 816.72 seconds with 63% accuracy. Moreover, this provides us with the ability to extend different classifiers and, therefore, image feature extraction is independent of classifying to optimize results.

6.2 Future Work

Future research can focus on optimizing features enhancement operation to obtain better results than shown in this work. This can provide many possibilities for further research to use incremental learning and deep learning features for better online classification results. It will be applicable in mobile and enhanced specifically for better user experience on a local domain and user-oriented space.

REFERENCES

1.13. Feature selection scikit-learn 0.20.0 documentation. (n.d.). Retrieved 2018-11-05, from http://scikit-learn.org/stable/modules/feature_selection.html

Akhi, A. B., Akter, F., Khatun, T., Uddin, M. S. (2018, July). Recognition and Classification of Fast Food Images. *Global Journal of Computer Science and Technology*. Retrieved 2018-12-30, from https://computerresearch.org/index.php/computer/article/view/1719

Bettadapura, V., Thomaz, E., Parnami, A., Abowd, G., Essa, I. (2015, January). Leveraging Context to Support Automated Food Recognition in Restaurants. *2015 IEEE Winter Conference on Applications of Computer Vision*, 580–587. Retrieved 2018-12-30, from http://arxiv.org/abs/1510.02078 (arXiv: 1510.02078) doi: 10.1109/WACV.2015.83

Blum, A. L., Langley, P. (1997, December). Selection of relevant features and examples in machine learning. *Artificial Intelligence*, *97*(1), 245–271. Retrieved 2019-07-28, from http://www.sciencedirect.com/science/article/pii/S0004370297000635 doi: 10.1016/S0004-3702(97)00063-5

Borisov, A., Eruhimov, V., Tuv, E. (2006). Tree-Based Ensembles with Dynamic Soft Feature Selection. In I. Guyon, M. Nikravesh, S. Gunn, L. A. Zadeh (Eds.), *Feature Extraction: Foundations and Applications* (pp. 359–374). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved 2019-07-28, from https://doi.org/10.1007/978-3-540-35488-8_16 doi: 10.1007/978-3-540-35488-8_16

Bossard, L., Guillaumin, M., Van Gool, L. (2014). Food-101 Mining Discriminative Components with Random Forests. In D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), *Computer Vision ECCV 2014* (Vol. 8694, pp. 446–461). Cham: Springer International Publishing. Retrieved 2018-12-29, from http://link.springer.com/10.1007/978-3-319-10599-4_29 doi: 10.1007/978-3-319-10599-4_29

Casimir, R., Boutleux, E., Clerc, G., Yahoui, A. (2006, March). The use of features selection and nearest neighbors rule for faults diagnostic in induction motors. *Engineering Applications of Artificial Intelligence*, *19*(2), 169–177. Retrieved 2019-07-28, from http://www.sciencedirect.com/science/article/pii/S0952197605000904 doi: 10.1016/j.engappai.2005.07.004

Cheng, W.-Y., Juang, C.-F. (2011, January). An incremental support vector machine-trained TS-type fuzzy system for online classification problems. *Fuzzy Sets and Systems*, *163*(1), 24–44. Retrieved 2019-01-02, from http://www.sciencedirect.com/science/article/pii/S0165011410003155 doi: 10.1016/j.fss.2010.08.006

Dean, J., Corrado, G., Monga, R., Chen, K., Devin, M., Mao, M., … Ng, A. Y. (2012). Large Scale Distributed Deep Networks. In F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1223–1231). Curran Associates, Inc. Retrieved 2019-02-06, from http://papers.nips.cc/paper/4687-large-scale-distributed-deep-networks.pdf

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, Li Fei-Fei. (2009, June). ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). Miami, FL: IEEE. Retrieved 2019-02-06, from http://ieeexplore.ieee.org/document/5206848/ doi: 10.1109/CVPR.2009.5206848

De Rosa, R., Orabona, F., Cesa-Bianchi, N. (2015, August). The ABACOC Algorithm: a Novel Approach for Nonparametric Classification of Data Streams. *arXiv:1508.04912 [cs, stat]*. Retrieved 2018-07-25, from http://arxiv.org/abs/1508.04912 (arXiv: 1508.04912)

Dodge, S., Karam, L. (2017, May). A Study and Comparison of Human and Deep Learning Recognition Performance Under Visual Distortions. *arXiv:1705.02498 [cs]*. Retrieved 2018-07-28, from http://arxiv.org/abs/1705.02498 (arXiv: 1705.02498)

Ege, T., Yanai, K. (2017, May). Simultaneous estimation of food categories and calories with multi-task CNN. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)* (pp. 198–201). doi: 10.23919/MVA.2017.7986835

Farinella, G. M., Moltisanti, M., Battiato, S. (2014, October). Classifying food images represented as Bag of Textons. In (pp. 5212–5216). IEEE. Retrieved 2018-08-11, from http://ieeexplore.ieee.org/document/7026055/ doi: 10.1109/ICIP.2014.7026055

Hall, M. A. (2000, May). *Correlation-based feature selection of discrete and numeric class machine learning* (Working Paper). University of Waikato, Department of Computer Science. Retrieved 2019-07-28, from https://researchcommons

.waikato.ac.nz/handle/10289/1024

He, K., Zhang, X., Ren, S., Sun, J. (2016, March). Identity Mappings in Deep Residual Networks. *arXiv:1603.05027 [cs]*. Retrieved 2018-12-29, from http://arxiv.org/abs/1603.05027 (arXiv: 1603.05027)

Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q. (2016, August). Densely Connected Convolutional Networks. *arXiv:1608.06993 [cs]*. Retrieved 2018-07-26, from http://arxiv.org/abs/1608.06993 (arXiv: 1608.06993)

Jeong, Y., Myaeng, S.-H. (2013, October). Feature Selection Using a Semantic Hierarchy for Event Recognition and Type Classification. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing* (pp. 136–144). Nagoya, Japan: Asian Federation of Natural Language Processing. Retrieved 2019-07-28, from https://www.aclweb.org/anthology/I13-1016

Kawano, Y., Yanai, K. (2013). Real-Time Mobile Food Recognition System. *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 1–7. doi: 10.1109/CVPRW.2013.5

Kawano, Y., Yanai, K. (2014). FoodCam: A Real-Time Mobile Food Recognition System Employing Fisher Vector. In D. Hutchison et al. (Eds.), *MultiMedia Modeling* (Vol. 8326, pp. 369–373). Cham: Springer International Publishing. Retrieved 2018-07-26, from http://link.springer.com/10.1007/978-3-319-04117-9_38 doi: 10.1007/978-3-319-04117-9_38

Kong, F., Tan, J. (2012, February). DietCam: Automatic dietary assessment with mobile camera phones. *Pervasive and Mobile Computing*, *8*(1), 147–163. Retrieved 2018-12-29, from http://www.sciencedirect.com/science/article/pii/S1574119211001131 doi: 10.1016/j.pmcj.2011.07.003

Krizhevsky, A., Sutskever, I., Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc. Retrieved 2019-02-06, from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., Yunsheng, M., . . . Hou, P. (2018, March). A New Deep Learning-Based Food Recognition System for Dietary Assessment on An Edge Computing Service Infrastructure. *IEEE Transactions* *on Services Computing*, *11*(2), 249–261. Retrieved 2018-12-24, from `https://ieeexplore.ieee.org/document/7837725/` doi: 10.1109/TSC.2017.2662008

Liu, H., Motoda, H. (2007). *Computational Methods of Feature Selection*. CRC Press. (Google-Books-ID: N1ViHNWZeQ0C)

Matsuda, Y., Hoashi, H., Yanai, K. (2012, July). Recognition of Multiple-Food Images by Detecting Candidate Regions. In (pp. 25–30). IEEE. Retrieved 2018-07-26, from `http://ieeexplore.ieee.org/document/6298369/` doi: 10.1109/ICME.2012.157

Ming, Z.-Y., Chen, J., Cao, Y., Forde, C., Ngo, C.-W., Chua, T. S. (2018). Food Photo Recognition for Dietary Tracking: System and Experiment. In K. Schoeffmann et al. (Eds.), *MultiMedia Modeling* (Vol. 10705, pp. 129–141). Cham: Springer International Publishing. Retrieved 2018-12-30, from `http://link.springer.com/10.1007/978-3-319-73600-6_12` doi: 10.1007/978-3-319-73600-6_12

Pan, L., Pouyanfar, S., Chen, H., Qin, J., Chen, S.-C. (2017, October). DeepFood: Automatic Multi-Class Classification of Food Ingredients Using Deep Learning. In *2017 IEEE 3rd International Conference on Collaboration and Internet Computing (CIC)* (pp. 181–189). San Jose, CA: IEEE. Retrieved 2018-12-24, from `http://ieeexplore.ieee.org/document/8181494/` doi: 10.1109/CIC.2017.00033

Polikar, R., Upda, L., Upda, S. S., Honavar, V. (2001, November). Learn++: an incremental learning algorithm for supervised neural networks. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *31*(4), 497–508. doi: 10.1109/5326.983933

Puhl, R. M., Heuer, C. A. (2009). The Stigma of Obesity: A Review and Update. *Obesity*, *17*(5), 941–964. Retrieved 2018-12-29, from `https://onlinelibrary.wiley.com/doi/abs/10.1038/oby.2008.636` doi: 10.1038/oby.2008.636

Rawat, W., Wang, Z. (2017, September). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, *29*(9), 2352–2449. Retrieved 2018-07-28, from `http://www.mitpressjournals.org/doi/abs/10.1162/neco_a_00990` doi: 10.1162/neco_a_00990

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., Lampert, C. H. (2017, July). iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5533–5542). Honolulu, HI: IEEE. Retrieved 2019-01-02, from http://ieeexplore.ieee.org/document/8100070/ doi: 10.1109/CVPR.2017.587

Rublee, E., Rabaud, V., Konolige, K., Bradski, G. (n.d.). ORB: an efficient alternative to SIFT or SURF. In *In ICCV.*

Sarwar, S. S., Ankit, A., Roy, K. (2017, December). Incremental Learning in Deep Convolutional Neural Networks Using Partial Network Sharing. *arXiv:1712.02719 [cs].* Retrieved 2018-11-05, from http://arxiv.org/abs/1712.02719 (arXiv: 1712.02719)

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y. (2013, December). OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks. *arXiv:1312.6229 [cs].* Retrieved 2019-02-06, from http://arxiv.org/abs/1312.6229 (arXiv: 1312.6229)

Simonyan, K., Zisserman, A. (2014, September). Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556 [cs].* Retrieved 2019-01-02, from http://arxiv.org/abs/1409.1556 (arXiv: 1409.1556)

Srivastava, R. K., Greff, K., Schmidhuber, J. (2015, May). Highway Networks. *arXiv:1505.00387 [cs].* Retrieved 2018-12-29, from http://arxiv.org/abs/1505.00387 (arXiv: 1505.00387)

Wagsta, K., Cardie, C., Rogers, S., Schroedl, S. (n.d.). Constrained K-means Clustering with Background Knowledge. , 8.

Yanai, K., Kawano, Y. (2015, June). Food image recognition using deep convolutional network with pre-training and fine-tuning. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)* (pp. 1–6). Turin, Italy: IEEE. Retrieved 2018-12-29, from http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7169816 doi: 10.1109/ICMEW.2015.7169816

Zagoruyko, S., Komodakis, N. (2016, May). Wide Residual Networks. *arXiv:1605.07146 [cs].* Retrieved 2018-12-29, from http://arxiv.org/abs/1605.07146 (arXiv: 1605.07146)

Zeiler, M. D., Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. In D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), *Computer Vision ECCV 2014* (Vol. 8689, pp. 818–833). Cham: Springer International Publishing. Retrieved 2019-02-06, from http://link.springer.com/10.1007/978-3-319-10590-1_53 doi: 10.1007/978-3-319-10590-1_53

Zhang, C., Fang, Z., Zhou, P., Pan, P., Cong, J. (2016). Caffeine: towards uniformed representation and acceleration for deep convolutional neural networks. In *Proceedings of the 35th International Conference on Computer-Aided Design - ICCAD '16* (pp. 1–8). Austin, Texas: ACM Press. Retrieved 2019-01-02, from http://dl.acm.org/citation.cfm?doid=2966986.2967011 doi: 10.1145/2966986.2967011