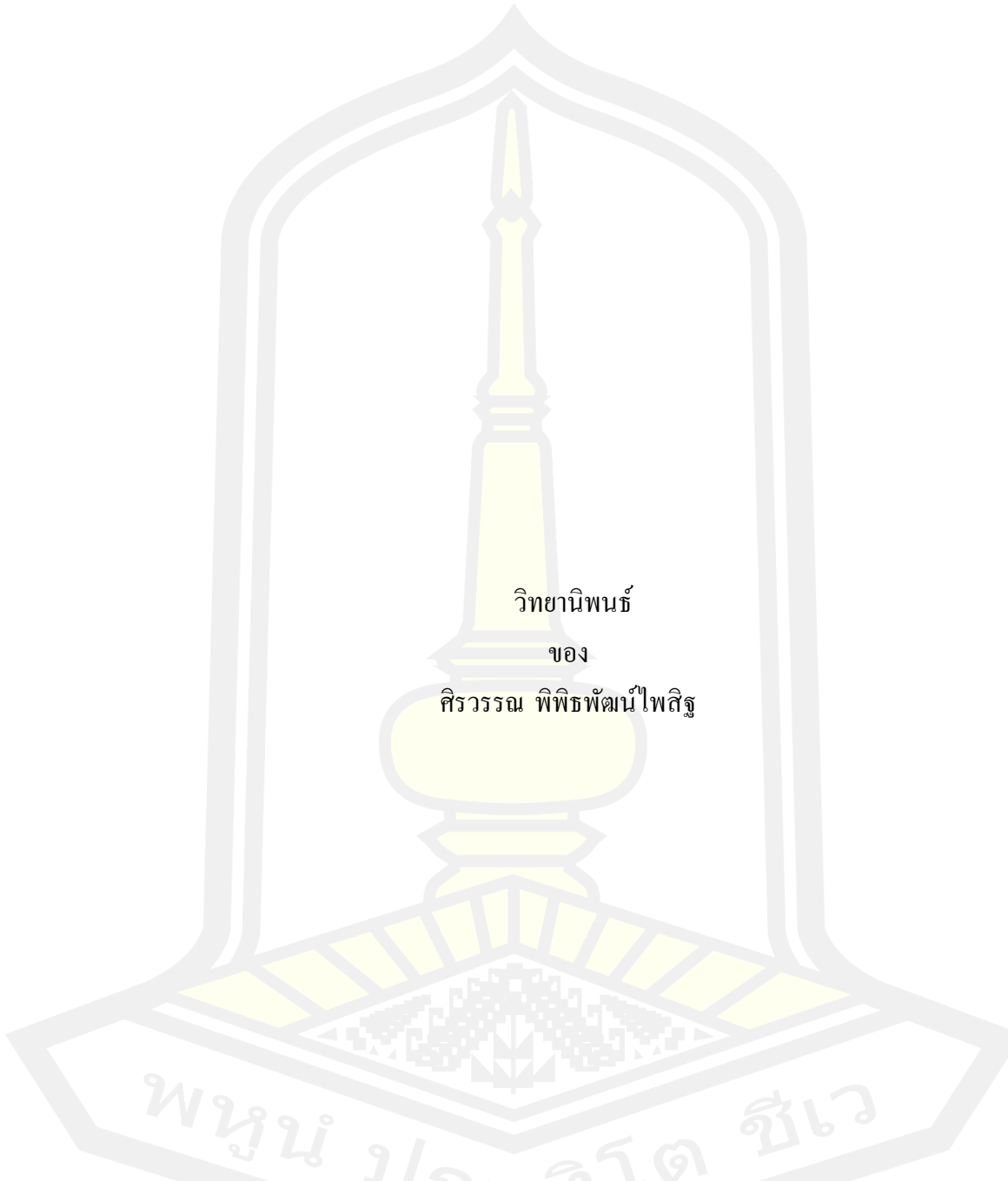
Deep Learning Approach for Food Image Recognition

Sirawan Phiphitphatphaisit

A Thesis Submitted in Partial Fulfillment of Requirements for
degree of Doctor of Philosophy in Information Technology
January 2022

กระบวนการเรียนรู้เชิงลึกสำหรับการรู้จำรูปภาพอาหาร



วิทยานิพนธ์

ของ

ศิรวรรณ พิพิธพัฒน์ไพสิฐ

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

มกราคม 2565

Deep Learning Approach for Food Image Recognition

Sirawan Phiphitphatphaisit

A Thesis Submitted in Partial Fulfillment of Requirements

for Doctor of Philosophy (Information Technology)

January 2022

The examining committee has unanimously approved this Thesis, submitted by Miss Sirawan Phiphitphatphaisit , as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology at Mahasarakham University

Examining Committee

..... Chairman
(Prof. Rapeepan Pitakaso , Ph.D.)

..... Advisor
(Asst. Prof. Olarik Surinta , Ph.D.)

..... Committee
(Asst. Prof. Rapeeporn Chamchong , Ph.D.)

..... Committee
(Asst. Prof. Chatklaw Jareanpon , Ph.D.)

..... Committee
(Asst. Prof. Phatthanaphong Chompoowises , Ph.D.)

Mahasarakham University has granted approval to accept this Thesis as a partial fulfillment of the requirements for the Doctor of Philosophy Information Technology

.....	.....
(Asst. Prof. Sasitorn Kaewman)	(Assoc. Prof. Krit Chaimoon , Ph.D.)
Dean of The Faculty of Informatics	Dean of Graduate School

TITLE Deep Learning Approach for Food Image Recognition

AUTHOR Sirawan Phiphitphatphaisit

ADVISORS Assistant Professor Olarik Surinta , Ph.D.

DEGREE Doctor of Philosophy **MAJOR** Information Technology

UNIVERSITY Mahasarakham University **YEAR** 2022

ABSTRACT

Food image recognition plays an important role in healthcare applications that monitor eating habits, dietary, nutrition, etc. Therefore, different deep learning approaches are proposed to address food image recognition. This dissertation presents three methods to deal with several challenges in recognizing food images.

Chapter 1 briefly introduces food image recognition systems and the research questions. Additionally, the objectives of the dissertation and contributions are described.

Chapter 2 proposed a new CNN model that modified MobileNetV1 architecture by decreasing the parameters but still achieved high accuracy. I replaced the average pooling layer and the fully connected layer (FC) with the global average pooling layer (GAP), followed by the batch normalization layer (BN) and rectified linear unit (ReLU) activation function. Moreover, I added the dropout layer to consider avoiding overfitting. The experimental results show that modified MobileNetV1 architecture significantly outperforms other architectures when the data augmentation techniques are combined.

Chapter 3 concentrated extracted robust features using the deep feature extraction technique. Firstly, I extracted the spatial features using CNN architectures. The spatial features were transferred into the Conv1D-LSTM network to extract the temporal feature. Finally, the deep features were classified using the softmax function. I presented six state-of-the-art CNN architectures, VGG16, VGG19, ResNet50, DenseNet201, MobileNetV1, and MobileNetV2, to extract the robust spatial features. The experimental results found that the ResNet50+Conv1D-LSTM network significantly outperformed other CNNs on the ETH food-101 dataset.

Chapter 4 presented an adaptive feature fusion network (ASTFF-Net) combining state-of-the-art CNN models and the LSTM network. Firstly, I extracted the spatial features using state-of-the-art ResNet50 architecture. Secondly, the temporal features were extracted using the LSTM network. Thirdly, the spatial-temporal features mapped to a similar resolution before concatenating. The experimental results showed that the ASTFF-Net achieved the best performances and outperformed other methods on Food11, UEC Food-100, UEC Food-256, and ETH

Food-101.

Chapter 5 comprises two main sections: the answers to the research questions and suggestions for future work. This chapter briefly explains the proposed approaches and answers three main research questions in food image recognition. Two main methods are planned and will be focused on in future work. The first is to reduce the training data size by applying the instance selection techniques to decrease computation time. The second is to focus on an instance segmentation technique that can segment and learn only at the exact food location, which will improve the performance of the food image recognition system.

Keyword : Food Image Recognition, Convolution Neural Network, Data Augmentation, Deep Feature Extraction Method, Long short-term memory, Adaptive Feature Fusion Technique, Spatial and Temporal Features

ACKNOWLEDGEMENTS

I would like to thank my esteemed supervisor – Assistant Professor Dr. Olarik Surinta for his invaluable supervision, support, and tutelage during the course of my Ph.D. degree. My gratitude extends to the Faculty of Business Administration and Information Technology, Rajamangkala University of Technology ISAN KhonKaen Campus for the funding opportunity to undertake my studies at the Faculty of Informatics, Mahasarakham University. Additionally, I would like to thank my friends, lab mates, colleagues, and research team for a cherished time spent together in the lab, and in social settings. My appreciation also goes out to my family and my daughter for their encouragement and support throughout my studies.

Sirawan Phiphitphatphaisit

TABLE OF CONTENTS

Page

List of Tables

Page

List of Figures

Page

Chapter 1
Introduction

According to the World Health Organization (WHO), worldwide obesity has nearly tripled, with more than 1.9 billion adults overweight; of these, over 650 million were obese (World Health Organization, 2018). Some 340 million children and adolescents aged 5 - 19 and 39 million children under the age of 5 were overweight or obese. Almost half of the children under five who were overweight lived in Asia, and obesity is now on the rise in low- and middle-income countries (World Health Organization, 2018). Obesity rates vary significantly by country as they are influenced by different lifestyles and diets. Southeast Asia has seen alarming increases in obesity rates within the past five years. Nauru has the highest obesity rate at 61.0%, while Vietnam has the lowest rate at 2.1%. Thailand has the 139th highest obesity rate globally of 36.2% out of a total population of 69 million (World Population Review, 2021). WHO used body mass index (BMI) to screen for overweight or obesity because it is a reliable indicator of body fatness. Still, it does not diagnose the health of an individual. Adults with a BMI greater than 25 and 30 indicate that they are overweight and obese, respectively.

Nowadays, overweight and obesity have become global problems that I have to be concern about. When I gain too much weight, I will have fat tissue that occurs as a consequence of consuming the extra calories in the diet. It is a common problem of dyslipidemia, cardiovascular diseases, and also increases the risk of diabetes. Moreover, other diseases might eventuate, such as escalating hypertension, respiratory problems and sleep disorder (Al-Abed, 2021). About 4.7 million people have died in 2017 because of obesity (Ritchie & Roser, 2017).

Many countries educate people to do daily physical activity and in awareness of nutrition and lifestyle. The nutritionists also recommend that people who are overweight should take care of themselves more than regular people. The best choice is to observe themself by recording their daily food intake and nutritional information each day in a process called called ‘food logging’. Also, people who observe themself

by recording their daily food intake lost more weight than people who do not do so. People can monitor what they eat and track their nutritional eating patterns (Butryn, Phelan, Hill, & Wing, 2007).

The rapid developments in smartphone technology provide an opportunity to develop healthcare applications that monitor eating habits, dietary, nutrition, etc. For example, Chaput, Klingenberg, Astrup, & Sjödin (2011) developed an application that monitors the eating and exercise habits of people. The application collects data relating to exercise using pulse monitors and performs analysis to provide advice on the health of users. Burke, Wang, & Sevick (2011) developed a weight management application that runs on a mobile platform. The personal information and daily food intake were recorded which helped a person to control their weight.

Due to the rapid development of artificial intelligence (AI) technology, many algorithms have been designed to recognize and calculate the daily food intake using food images. The application developer also invented food recognition systems that involved AI and other technologies. It allows people to manage their food consumption behavior themselves. Moreover, the deep learning approach has become more popular and has been proposed to address food image recognition. I will briefly introduce the food image recognition systems as follows.

1.1 Food Image Recognition Systems

Food image recognition systems using the deep learning method are successful methods to help people track their dietary habits based on real-world food images. Indeed, deep learning to extract the information from the food images is a relatively low-cost and robust method. However, it is extremely challenging to extract information from real-world food images because people can take photos in different styles and sometimes several objects appear in the photo, not just the food.

Krizhevsky et al. (2012) proposed the convolutional neural network (CNN) architecture, a type of deep learning method, namely AlexNet, to address many problems in image recognition systems. Consequently, various CNN architectures were invented, such as VGGNet, GoogLeNet, ResNet, and DenseNet (K. He, Zhang, Ren, & J., 2016; Huang, Liu, Van Der Maaten, & Weinberger, 2017; Simonyan &

Zisserman, 2014; Szegedy et al., 2015). Currently, the CNN method is widely employed for image recognition problems.

Although, the CNN methods require a large amount of training data to create a robust model. The food images are usually downloaded from social media or the internet; such as the Food-101 dataset, the well-known food image dataset, which collected all the images from the foodspoting.com website. It contains more than 100,000 images of 101 food categories (Bossard & Gool, 2014). Further, the image processing methods such as adjustment and transformation are proposed to reduce the noise from resolution inconsistency and nonuniform illumination (Jiang, Qiu, Liu, Huang, & Lin, 2020; Ng, Xue, Wang, & Qi, 2019; B. T. Nguyen, Dang-Nguyen, Tien, Phat, & Gurrin, 2018; Park et al., 2019).

When the food images are insufficient, I can also perform data augmentation techniques, such as random cropping, rotation, and flipping, to enlarge the number of images used while training to create the deep learning models (J. He et al., 2021; Jiang et al., 2020; Ng et al., 2019; Sahoo et al., 2019). Hence, all images are divided into training, validation, and test sets to create a robust model. In this process, the robust model is derived by tuning the deep learning hyperparameters.

The following section describes the CNN architecture in detail with three main parts, including CNNs, CNNs for food image recognition, and deep feature extraction for food image recognition.

1.1.1 Convolutional Neural Networks

Deep learning is a type of machine learning based on artificial neural networks (Hinton, 2009). The deep learning algorithm is designed to solve complex classification problems, such as image recognition, language translation, and speech recognition (Fayyaz & Ayaz, 2019; Haque, Verma, Alex, & Venkatesan, 2020; Kesav & Jibukumar, 2021). However, it requires high-performance hardware because it involves several complex mathematical calculations that compute from a large amount of data (McAllister, Zheng, Bond, & Moorhead, 2018). The popular deep learning algorithms are convolutional neural networks (CNNs), long short-term memory networks (LSTMs), recurrent neural networks (RNNs), and deep belief networks (DBNs) (Shrestha & Mahmood, 2019).

In this section, I mainly focus on CNN architectures. LeCun et al. (1989) proposed the first CNN architecture to recognize handwritten digit recognition, called LeNet-5. The architecture of LeNet-5 is shown in Figure 1. The figure shows that LeNet-5 consists of five layers, including 1) a convolution layer with six feature maps, 2) a pooling layer with six feature maps, 3) a convolution layer with 16 layers, 4) a pooling layer with 16 layers, and 5) a fully connected layer with the size of 120, 80, and 10, respectively. Hence, the basic convolution operations are described in the following section.

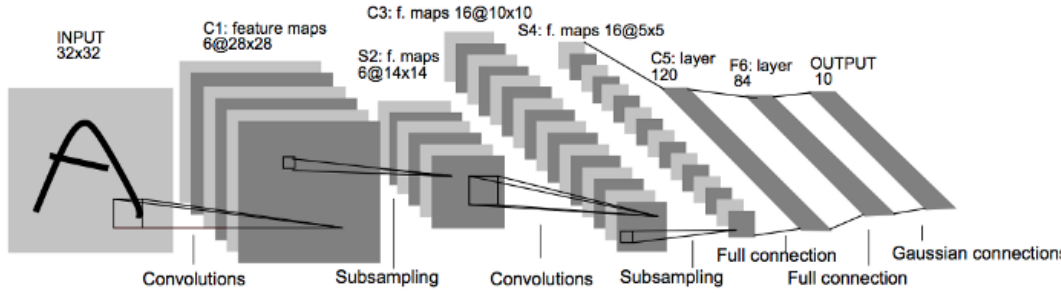


Figure 1 *Illustration of LeNet-5 architecture (Y. A. LeCun, Kavukcuoglu, & Farabet, 2010)*

1.1.1.1 Convolution Layer

A convolution layer is the main building block of a CNN architecture and is proposed to extract robust features from images (Y. LeCun, Bottou, Bengio, & Haffner, 1998). The operation of the convolution layer was designed to be similar to the convolution method in computer vision. The tiny kernel sizes, such as 3x3, 5x5, and 7x7, are aimed to calculate with the original image to create the new feature map. It computes the kernel over the original image from the top left until the right button regions. The convolution operation is calculated by multiplying the corresponding values from the original image and kernel and adding them together. An illustration of the convolution layer is shown in Figure 2.

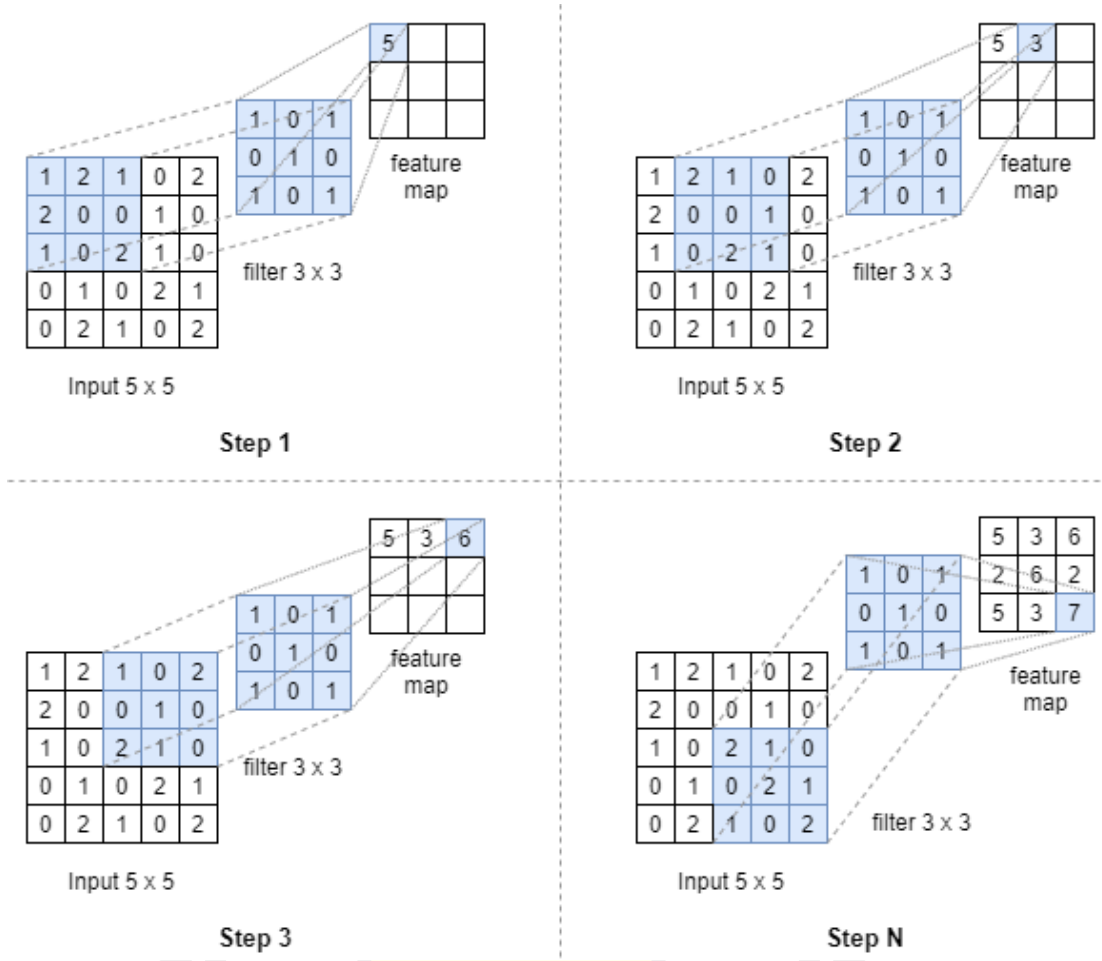


Figure 2 *Examples of the convolution operation. The hyperparameters used in the example are a filter size of 3 x 3, no padding, and a stride of 1.*

However, the size of the original image and the feature map do not show equal size after applying the convolution operation (see Figure 2). In this case, I proposed to use the padding operation when producing and image of equal size as the original image is required. The padding operation is shown in Figure 3.

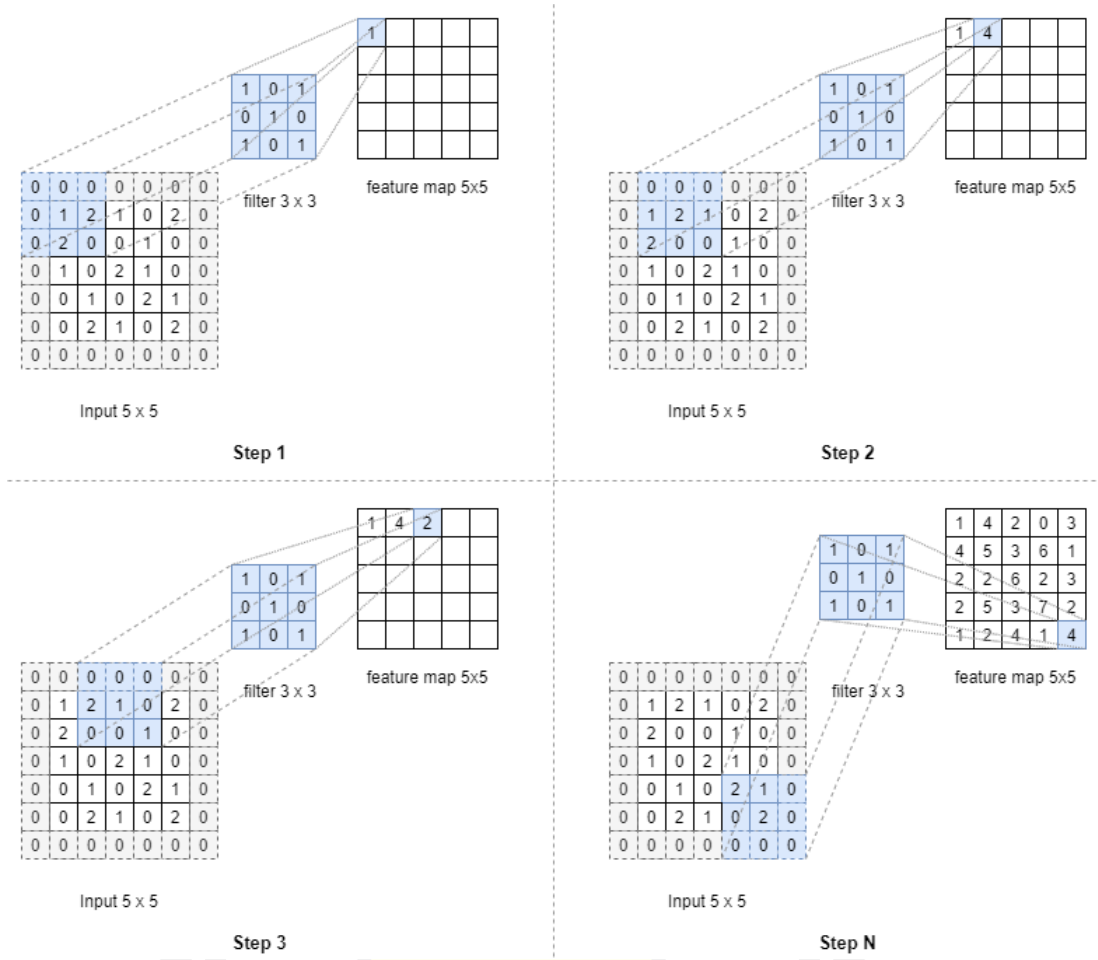


Figure 3 *Illustration of the convolution operation by adding the padding operation.*

1.1.1.2 Pooling Layer

Pooling layers are designed to downsample the dimensionality of the feature maps and decrease the number of learnable parameters (Boureau, Ponce, & LeCun, 2010). The pooling layer is usually attached to the network after the convolution layer. It usually speeds up computation and makes features more robust. The pooling operation requires a 2D filter slide above feature maps and calculating the features, such as maximum and average pixel values within the region are covered by the 2D filter. The traditional and popular pooling layers are max and average pooling layers. In the max pooling layer, the maximum value in each pool is chosen as the representative. While with average pooling, the average value in each pool is chosen. Examples of the pooling layers are shown in Figure 4.

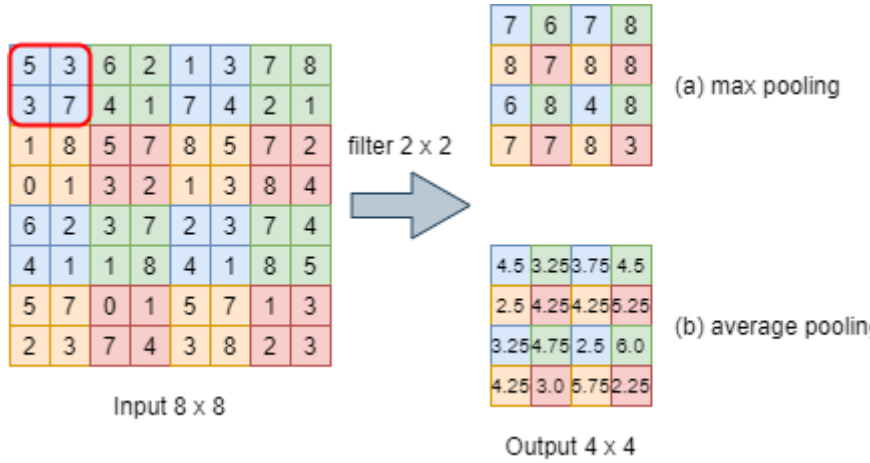


Figure 4 *Examples of (a) max pooling layer and (b) average pooling layers with a filter size of 2x2, no padding, and a stride of 2.*

The global average pooling (GAP) layer was invented to minimize the parameters of a 3D feature map in the CNN model (Lin, Chen, & Yan, 2014). The pixel values of each region in each feature map are averages and represented as the vector. The GAP layer is proposed to replace the flattened layers (see Figure 5b). Then the vector generated by the GAP layer is transferred immediately toward the softmax layer. The operation of the GAP layer is shown in Figure 5a.

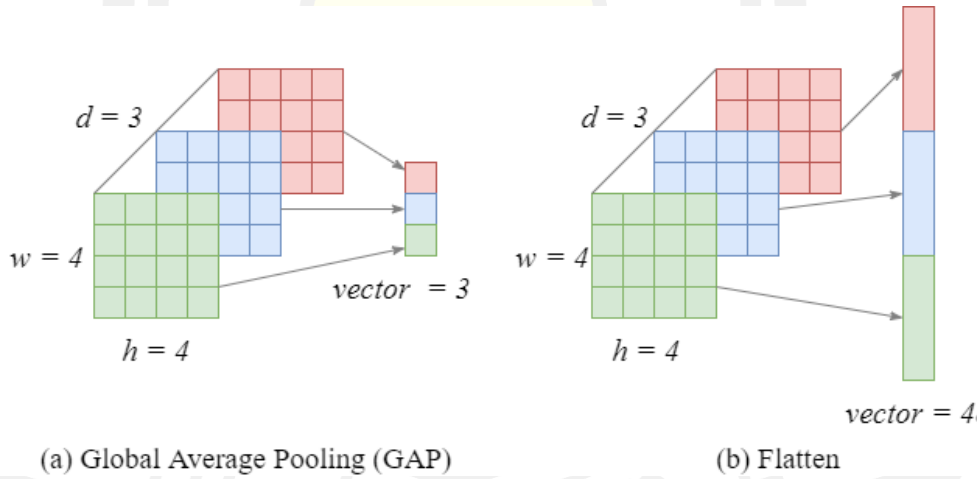


Figure 5 *Illustration of the (a) GAP layer and (b) flatten layer with hyperparameters of width (w) = 4, height (h) = 4, dimension (d) = 3. The output vector of the GAP layer is the only vector of 3 values and the flatten layer is the vector of 16 values.*

1.1.1.3 Fully Connected Layer

The fully connected (FC) layer (Y. LeCun et al., 1998) was designed to connect all the neurons between two different layers; the previous and

current layers. The FC layer is permanently attached at the end of the CNN architecture and combined with an activation function, such as softmax and sigmoid functions, to perform the output of CNN networks (Nwankpa, Ijomah, Gachagan, & Marshall, 2018) The outputs of the CNN model are probabilities that are employed to predict the objects.

1.1.2 CNNs for Food Image Recognition

For food image recognition, many CNN architectures have been proposed to recognize food images. Hassannejad et al. (2016) presented a 54-layer network to classify food images. It achieved an accuracy of 88.28%, 81.45%, and 76.17% in Food-101, UEC-FOOD100, and UEC-FOOD256, respectively. Liu et al. (2016) invented the DeepFood network, which modified the Inception module using a 1×1 convolutional kernel to reduce the input size and feed it to the next layer. The DeepFood network obtained an accuracy of 76.30% on UEC-FOOD100 and 54.70% on UEC-FOOD256.

Subsequently, Aguilar et al. (2017b) presented a Fusion CNN method that combined state-of-the-art CNN architectures; ResNet and Inception. The Fusion CNN model achieved an accuracy of 86.71% on the Food-101 dataset and 72.12% on the Food-11. Pandey et al. (2017) proposed the ensemble CNN network, including ResNet, AlexNet, and GoogLeNet. The accuracy of 72.12% was achieved from the ensemble CNN network.

1.1.3 Deep Feature Extraction for Food Image Recognition

CNN architecture contains two main components; feature extraction and classification. Many state-of-the-art CNN architectures, such as AlexNet, VGG-16, GoogLeNet, have been proposed to extract the robust feature, called the deep feature method (Sengur, Akbulut, & Budak, 2019; Zheng, Zou, & Wang, 2018). Therefore, the deep features can be sent to the machine learning techniques, such as support vector machine (SVM) and random forest, to create a model and recognize the food images. Ragusa et al. (2016) proposed to use the VGG-S, Network-in-Network, and AlexNet, to extract deep features and then train with the SVM method. The experimental results showed that the VGG-S combined with the SVM method achieved an accuracy of 92.47%, the VGG-S and the Network-in-Network achieved

an accuracy of 90.82% and 84.95%, respectively. In addition, Farooq and Sazonov (2017) extracted the deep features using AlexNet architecture. The deep features were extracted from layers 6, 7, and 8. The deep features of each layer were fed to the linear SVM method for recognition. The result showed that the extracted deep features from layer 6 obtained the highest accuracy of 94.01% on the PFID dataset.

1.2 Research Aim

This research aimed to design novel deep learning methods to improve the performance of food image recognition systems.

1.3 Research Questions and Research Studies

The main research question that motivates this dissertation is: How can I enhance the performance of the food image recognition system using the deep learning method? This dissertation proposes to contribute novel solutions to deal with the problems of food image recognition. I address the following research questions:

RQ1: Training the model with deep learning methods such as convolutional neural network (CNN) typically requires a large amount of training data to create an effective model (Russakovsky et al., 2015). The benchmark food image datasets, such as the ETH food-101, contain 101,000 real-world food images (Bossard & Gool, 2014). Indeed, the CNN architectures spent expensive training time to create the effective CNN models. Is it possible to decrease the size of the training data but still provide the same performance of recognition?

To find out the answer, I will focus on modifying a state-of-the-art lightweight CNN model. The hyperparameters and computational layers of the CNN model are also considered. Moreover, I will consider the data augmentation techniques that benefit learning to build an effective CNN model from distinctive food images. Will these methods encourage improved performance of food image recognition systems?

RQ2: In computer vision, hand-crafted feature techniques are presented to extract the specific information existing in the image. Indeed, it mainly focuses on extracting local features. The well-known hand-crafted feature techniques, include local binary pattern (LBP) (Ojala, Pietikainen, & Harwood, 1994), histogram of

oriented gradient (HOG) (Dalal & Triggs, 2005), scale-invariant feature transform (SIFT) (Lowe, 2004), and speeded up robust features (SURF) (Bay, Ess, Tuytelaars, & Van Gool, 2008). Nowadays, the CNN technique is a capable technique that includes feature extraction and recognition. As for the feature extraction, the CNN can extract robust special features, including low-level and high-level features, called the deep feature method (Y. Chen, Jiang, Li, Jia, & Ghamisi, 2016; Paul et al., 2016). Is this a potential approach to manipulate real-world food images that also have many categories? If possible, I will then be interested in using state-of-the-art CNN architecture to extract the deep features and enhance the food image recognition system.

RQ3: The deep feature extraction method always provides robust features and guarantees high accuracy performance on the real-world food image dataset (Phiphitphatphaisit & Surinta, 2021). Is there any approach that will prevent the deep feature extraction method using Conv1D and LSTM networks?

In order to answer all these research questions, Chapter 2 to Chapter 4 describe the research that succeeded. I will present concrete solutions to these research questions in Chapter 5.

1.4 Contributions

The contribution of the dissertation is a novel deep learning technique to extract the robust features and provide the best performance for food image recognition systems. The work reported in this dissertation involved experiments on four real-world food image datasets containing Food-11, UEC Food-100, UEC Food-256, and ETH Food-101. The contributions of the dissertation are as follows.

In chapter 2, I modified the state-of-the-art lightweight MobileNetV1, called modified MobileNetV1. In this approach, I eliminated the two last layers; the average pooling layer and fully connected layer (FC), and then attached three new layers into the MobileNetV1 architecture, which were; the global average pooling layer (GAP), the batch normalization layer (BN), and rectified linear unit (ReLU) activation function. Additionally, data augmentation techniques were proposed to address the

problem when amount of training data was decreased, consisting of rescaling, rotation, width shift, height shift, horizontal flip, shear, zoom, and random cropping. This chapter is based on the following publication.-

Phiphiphatphaisit, S., & Surinta, O. (2020). Food Image Classification with Improved MobileNet Architecture and Data Augmentation. In The 3rd International Conference on Information Science and Systems (ICISS), pages 51–56. ACM.

In chapter 3, the main focus is extracting the powerful features using the deep feature extraction technique. I extracted the spatial features with state-of-the-art convolutional neural network (CNN) architectures. Subsequently, the spatial features were given to the convolutional 1D (Conv1D), followed by the LSTM network to compute and extract the temporal feature, called Conv1D-LSTM. Therefore, I decrease the dimensionality of the feature maps before classifying the food images using the global average pooling (GAP) layer. The content of this chapter is based on the following publication.-

Phiphitphatphaisit, S., & Surinta, O. (2021). Deep feature extraction technique based on Conv1D and LSTM network for food image recognition. Engineering and Applied Science Research, 48(5), pages 581-592.

Finally, Chapter 4 proposes the adaptive feature fusion network, called ASTFF-Net, to improve the accuracy of the food image recognition systems. The proposed ASTFF-Net was the combination between state-of-the-art CNN models and the LSTM network. The ASTFF-Net is closely related to the Conv1D-LSTM. However, the Conv1D-LSTM network was created as a sequential model, while the ASTFF-Net was designed to connect the deep features extracted from CNN and LSTM networks by applying a concatenation operation. I achieved high accuracies on real-world food image datasets; Food11, UEC Food-100, UEC Food-256, and ETH Food-101.

Chapter 2
Deep Learning Techniques

The real-world food image is a challenging problem for food image classification, because food images can be captured from different perspective and patterns. Also, many objects can appear in the image, not just foods. To recognize food images, in this chapter, I propose a modified MobileNetV1 architecture that is applies the global average pooling layers to avoid overfitting the food images, batch normalization, rectified linear unit, dropout layers, and the last layer is softmax. The state-of-the-art and the proposed MobileNetV1 architectures are trained according to the fine-tuned model. The experimental results show that the proposed version of the MobileNetV1 architecture achieves significantly higher accuracies than the original MobileNetV1 architecture. The proposed MobileNetV1 architecture significantly outperforms other architectures when the data augmentation techniques are combined.

2.1 Introduction

Nowadays, people are becoming obese and overweight due to the imbalance between calorific intake and use. This increases the risk of other diseases such as diabetes, sleep apnea, acid reflux, and heart disease (Must et al., 1999). Nutritionists advise obese and overweight people to exercise and to monitor their daily consumption of calories (Fatehah, Poh, Shanita, & Wong, 2018). Due to the assessment of calorie intakes into the body, Ege and Yani (2017) proposed a multi-task convolutional neural network (CNN) method that allows the CNN architecture to learn from food calories, categories, ingredients, and cooking directions data. Furthermore, Myers et al. (2015) presented a system that recognizes the contents of food from a single image, and then predict calories using the CNN based classifier. Then, people can estimate calories from food images.

In recent years, most research in food image classification has focused on hand-crafted features that consist of a color histogram (Martinel, Piciarelli, & Micheloni, 2016; Yanai & Kawano, 2015), local binary pattern (LBP) (Martinel et al., 2016; D. T. Nguyen, Zong, Ogunbona, Probst, & Li, 2014), scale invariant feature transform (SIFT) (Martinel et al., 2016), histogram of oriented gradients (HOG)

(Martinel et al., 2016; Yanai & Kawano, 2015), and speeded up robust feature (SURF) (Bossard & Gool, 2014). These hand-crafted methods are combined with machine learning algorithms to classify food images. Due to the large-scale of food image datasets, researchers proposed to use deep learning algorithms to learn from the large-scale food image dataset such as the ETH Food-101 dataset which contains 101,000 images from 101 food categories; Food-256 dataset, a data set of 256 food categories with approximately 32,000 food images (Bossard & Gool, 2014; Hassannejad et al., 2016; Kawano & Yanai, 2014b). Yanai and Kawano (2015) used a pre-trained model of AlexNet architecture for the feature extraction method. This method extracts 6,144 features from the image. In Hassannejad et al. (2016), the data augmentation techniques consist of brightness, contrast, saturation, and hue and are applied to food images before feeding to the Inception V3 network. Ming et al. (2018) proposed the DietLens, which is a prototype of tracking dietary intake system for Singapore hawker food. The core architecture of the DietLens is the ResNet-50, which contains 50 convolutional layers and one fully connected layer and experiments on 87,470 images. The FoodNet (Pandey et al., 2017), which is an ensemble deep neural network, is proposed to classify the ETH Food-101 dataset. This network combined three well-known networks (AlexNet, GoogLeNet, and ResNet) as the ensemble network. The output of three networks and concatenate are passed to a fully connected layer to classify food images.

Figure 6 *Example of ETH Food-101 dataset. a) The apple pie category and b) the similarity shape between two categories of apple pie (first row) and Baklava (second row).*

The challenge of food image classification is that food images from the same category are captured with different patterns, shapes, and perspective, accordingly to the people who take the image. For example, there are many objects such as forks and

spoons, glasses, and bottles that appear in the image. For example of ETH Food-101 dataset, has many different apple pie images (that include other objects, patterns, shapes, and scenes) that appear in the apple pie category, as shown in Figure 6a). Even the similarity shape and pattern between the two categories of apple pie and Baklava, as shown in Figure 6b). These kinds of images can decrease the performance of the food image classification.

Contributions: In this research, the main contribution is the use of the state-of-the-art deep convolutional neural network, called MobileNetV1 architecture and our modified MobileNetV1 architecture is applied to recognize a challenging ETH Food image dataset that contains 101 food categories. In our modified version, I reduce the number of parameters in the model by replacing the average pooling with the global average pooling (GAP) layers; then the overfitting is decreased. Subsequently, the batch normalization (BN), rectified linear unit (ReLU), and dropout layers, are utilized instead of the fully connected layers. Finally, the softmax layer is calculated. The results show that the modified MobileNetV1 architecture outperforms when compared to the original MobileNetV1 architecture. Moreover, I evaluate most effective data augmentation techniques to random creating images in the ETH Food-101 dataset. I compared data augmentations and combined with the cropping image before passing to train the model. Also, the accuracy increased by approximately 5%. Finally, the modified MobileNetV1 architecture when combined with the data augmentation techniques outperforms the other methods.

Outline of the chapter. The chapter is organized as follows. Section 2.2 briefly explains machine learning methods in food image classification. In section 2.3, the MobileNetV1 and the modified MobileNetV1 architecture are explained. In section 2.4, the data augmentation techniques are presented. Experimental results are reported in section 2.5. The last section is the conclusion and future work.

2.2 Related Work

Hand-crafted feature extraction methods (Nanni, Ghidoni, & Brahnam, 2017) are used in many image classification applications. In D. T. Nguyen et al. (2014), two feature extraction methods consisting of a non-redundant local binary pattern (NRLBP) and the shape context descriptor of the interest points, called structure

information are used to describe the local appearance information of food images. The achieved accuracy shows that the combination of the two features can improve classification performance. In Yanai and Kawano (2015), the first step uses, the color patches and RootHOG patches, (which is a square root of the L1 normalized HOG) to extract the data from the images. In the second step, the information from the first step is sent to a Fisher vector to encoding and used as the feature vector. This method achieved an accuracy of 65.3% on the UEC Food-100 dataset. In addition, Martinel et al. (2016) presented the supervised extreme learning committee approach (ELM) to learning attributes of color, shape, texture, and local features. Then, the output of the ELMs is fed into the structured support vector machine (SVM) to classify food images. The performance achieved by this method is 55.89% and 84.34% on ETH Food-101 and UEC Food-100, respectively.

Nowadays, convolutional neural networks (CNNs), which are the most successful, and widely used for image classification problems (Russakovsky et al., 2015). Although, many CNN architectures can compute due to the large-scale images (Russakovsky et al., 2015) and obtain very high accuracy (C. Liu et al., 2018; Ming et al., 2018). In the area of food image classification, state-of-the-art CNN architectures such as AlexNet, GoogLeNet, and ResNet are proposed (Pandey et al., 2017), although, the experimental results obtained with them did not obtain high accuracy. Pandey et al. (2017) invented a CNN-based ensemble network, called FoodNet architecture. This architecture consists of a fine-tuned model of AlexNet, GoogLeNet, and ResNet. The networks compute feature vectors and then concatenate all of the feature vectors, and a rectified linear unit (ReLU) used as a non-linear activation. Then, data is passed to a fully connected layer and the softmax function used to predict the output of the food image. The experiments showed that the FoodNet architecture obtained the Top-1 accuracy of 72.12% on ETH Food-101 and 73.50% on Indian food database. Also, the result was not good when the feature vector from the FoodNet architecture was fed into the SVM classifier.

As for the pre-trained model, In Yanai and Kawano (2015), the fine-tuning of the deep CNN pre-trained model based on AlexNet network, called DCNN was proposed to examine three food image datasets. The results showed that the fine-tuned DCNN achieved the Top-1 accuracy of 78.77%, 67.57%, and 70.40% on UEC Food-

100, UEC Food-256, and ETH Food-101 datasets, respectively. The Inception networks (Hassannejad et al., 2016; C. Liu et al., 2016) are proposed to address the food image classification. Lin et al. (2016) presented the DeepFood network to recognize the food image for computer-aided dietary assessment. The DeepFood network, which is applied to an Inception module by adding 1x1 convolutional layers and then connected with two inception modules via an additional max-pooling layer. The best Top-1 accuracy results on UEC Food-256, UEC Food-100, and ETH Food-101 were 54.7%, 76.3%, and 77.4%, respectively. Hassannejad et al. (2016) invented a deep network with 54 layers based on Inception V3 to classify three well-known food image datasets and achieved 88.28% on ETH Food-101, 81.45% on UEC Food-100, and 76.17% on UEC Food-256 datasets as top-1 accuracy.

Additionally, data augmentation is proposed to address the problem of insufficient data and to increase the performance of the image classification (Attokaren, Fernandes, Sriram, Murthy, & Koolagudi, 2017; Yunus et al., 2019). The data augmentation is also widely used in plant (Pawara, Okafor, & Schomaker, 2017) and animal (Okafor, Schomaker, & Wiering, 2018), and food (Yunus et al., 2019) image recognition.

2.3 MobileNetV1 Architecture

I used MobileNetV1 architecture presented by Howard et al. (2017) that is designed and based on depthwise separable convolutions to build a lightweight deep CNN that makes a model too small and reduces the computation time. The diagram in Figure 7a) illustrates the MobileNetV1 architecture. Consequently, MobileNetV1 can be implemented for several recognition problems such as object detection, face attributes, fine-grain classification, and landmark recognition.

1.3.1 Our Modified MobileNetV1 Architecture

Our modified MobileNetV1 architecture was as follows. First, I used the pre-trained model of MobileNetV1 architecture. I decided to remove three layers, including the average pooling, fully connected, and softmax layers from the original network. Second, three extra layers; the global average pooling (GAP) layers, the batch normalization (BN), and softmax layers are attached. The main objective of our modified MobileNetV1 architecture is helping the network to train faster and

achieving higher accuracy. Then, the dropout method is proposed to prevent overfitting. Also, the batch normalization layer helps the network to train faster. The activation function called the rectified linear unit (ReLU) is computed between the batch normalization layer and the dropout layer. After I applied the GAP layers instead of the average pooling, it shows that the parameters in the model are decreased, and impact directly on the size of the model. Finally, for training the proposed network, I used the fine-tuned MobileNetV1 to train the network on the ETH Food-101 dataset. The modified MobileNetV1 architecture as shown in Figure 7b).

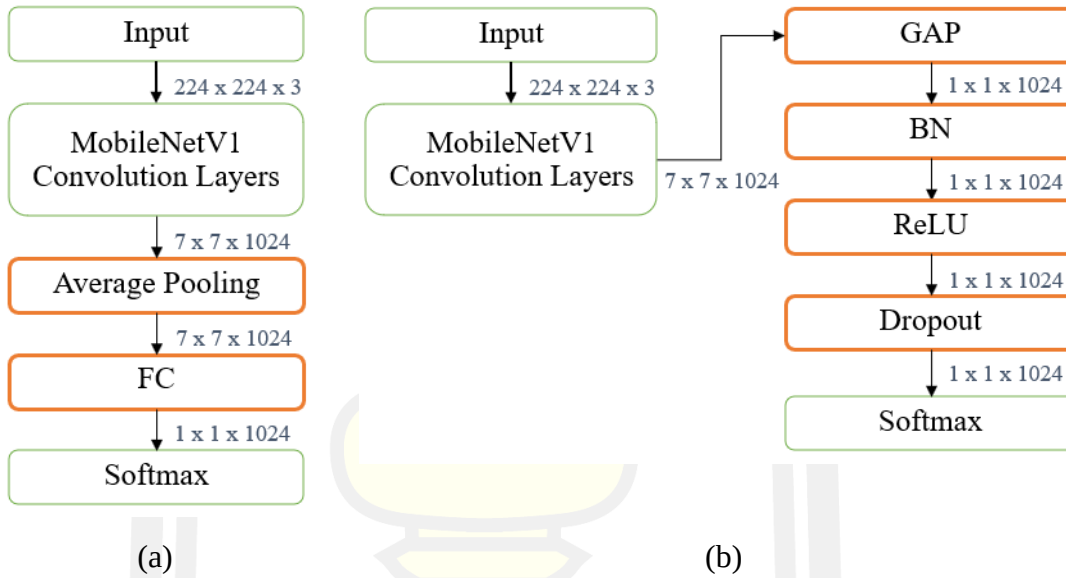


Figure 7 *The architectures of the MobileNetV1. (a) the original MobileNetV1 and, (b) the modified MobileNetV1 architectures.*

1.3.2 Depthwise Separable Convolutions

The MobileNetV1 architecture is computed based on depthwise separable convolutions (DS). The concept of decomposition of convolution called factorization is considered to factorize a standard convolution into a depthwise convolution. After that, all depthwise convolution layers are computed with 1x1 convolution called a pointwise convolution, and then combined as the outputs to the next layer. The diagram in Figure 7a) shows the detail of the MobileNetV1 that includes convolutional, depthwise separable convolutions (DS), average pooling, fully connected (FC), and softmax layers. Figure 7b) shows an in-depth explanation of the

DS layer consisting of depthwise convolution, batch normalization (BN), and rectified linear unit (ReLU), respectively.

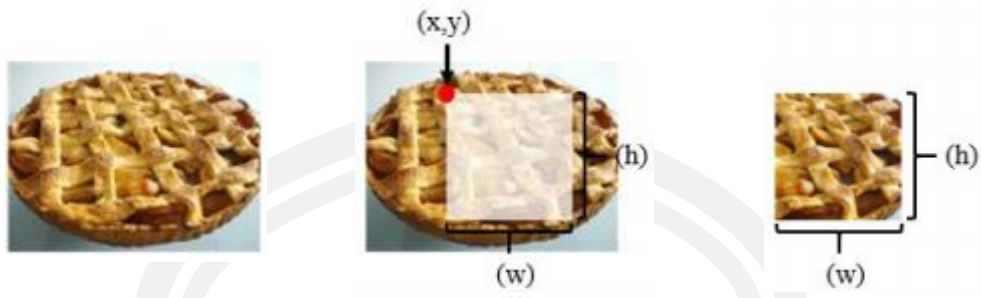
2.4 Data Augmentation Techniques

Data augmentation is a technique to generate new training image data that relate to the same image. Many data augmentation techniques such as rotation, horizontal, vertical, flip, width shift, height shift techniques are applied to the image recognition problems and the accuracy performance is improved (Yunus et al., 2019). Samples of image augmentation are shown in Figure 8. In this thesis, the data augmentation techniques applied to our experiments consists of rescaling, rotation, width shift, height shift, horizontal flip, shear, and zoom.

(a) (b) (c) (d) (e)

Figure 8 *Example of the data augmentation images: (a) original, (b) rotation, (c) width shift, (d) height shift, and (e) horizontal flip images.*

Additionally, the image randomly changes to generate a new image in each training epoch, according to the range of the parameters. Furthermore, random cropping is applied (Takahashi, Matsubara, & Uehara, 2020). In this method, the position of points (x, y) are random, then it automatically crops and resizes to the target size, as shown in Figure 9. In this experiment, the size of the image is 224x224 pixel dimension.



(a) (b) (c)

Figure 9 *Illustration of the random cropping method. (a) Original food image, (b) random points* (x, y) *and crop sizes of the cropped image* (w, h)*, and (c) the random cropping image used in training process.*

2.5 Experimental Setup and Results

2.5.1 ETH Food-101 dataset

In this study, I evaluate the deep CNN architectures on the benchmark food image dataset. The real-world food images were collected by downloading from foodspotting.com website. The food images are a mix of eastern and western meals such as apple pie, hamburger, sashimi, ramen, peking duck. The challenging dataset consists of 101,000 food images from 101 food categories, called the ETH Food-101 dataset (Bossard & Gool, 2014). Examples of the food images are shown in Figure 10.

Figure 10 *Sample real-world food images from the ETH Food-101 dataset.*

1.5.2 Experimental setup

Due to the large number of images in the dataset, I divided it into four subsets (Set I, Set II, Set III, and Set IV) sizes of 10,100 (randomly selected 100

images from each category), 20,200, 30,300, and 40,400 images to perform all of the experiments. Images in each subset were divided into training, validation, and testing sets of 70%, 10%, and 20%, respectively. For the training of the deep CNN architectures, I used the transfer learning with the following parameter settings: stochastic gradient descent (SGD) solver, batch size of 16, learning rate at 0.0001. I note that entire experiments were carried out using the TensorFlow platform running on Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz, 8GB RAM.

In the experiments, firstly, I used the original food images from the ETH Food-101 dataset to experimented with the MobileNetV1 architectures in order to find the appropriate training epoch. Secondly, the first data augmentation called random cropping was employed. The program randomly cropped from a part of a food image and resize to the target size, which was 224x224 pixel dimension. Thirdly, the data augmentation techniques consisted of rescaling, rotation, width shift, height shift, horizontal flip, shear, and zoom applied according to the random parameters. Suddenly, the food images randomly change in each training epoch. Finally, the random cropping image and the data augmentation techniques are combined.

1.5.3 Experimental results

I used the accuracy and standard deviation to evaluate the performance of the deep CNN architectures on ETH Food-101 dataset. From the first experiment, it is essential to indicate that a huge number of food images can increase recognition performance. I set up the number of training to 50 epochs, which is similar to previous reports (Attokaren et al., 2017; Pandey et al., 2017; Zheng et al., 2018). The accuracy of Set I with 10,100 images and Set IV with 40,400 images were significantly different. The accuracy results improved from around 42% to 57% when testted on the original MobileNetV1 architecture. Moreover, the results improve from 46% to 67% when performed on the modified MobileNetV1 architecture, when accuracy increased by more than 10%, as shown in Figure 11. This clearly indicates that recognition performance is increased when using more food images. However, I found that modified MobileNetV1 will decrease number of parameter and testing time around 24% and 7.5%, respectively, as show in Table 1.

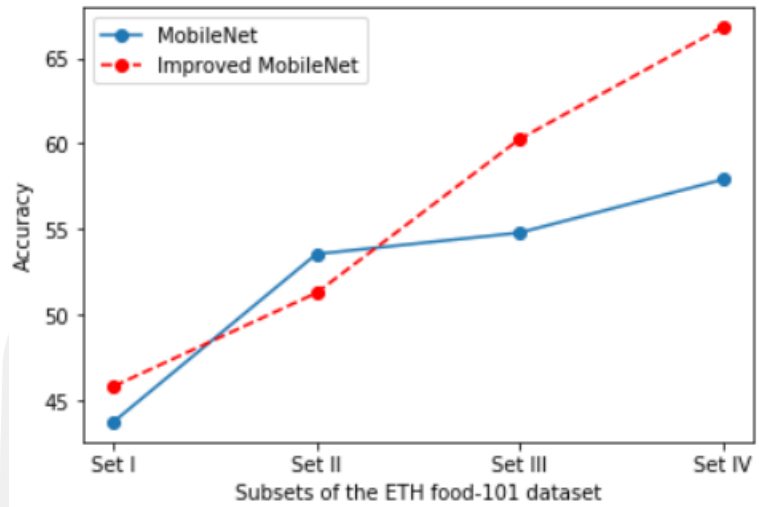


Figure 11 *The performance of the MobileNetV1 and modified MobileNetV1 architectures versus the different number of training samples (Set I – Set IV) on the ETH Food-101 dataset.*

Table 1 *The performance results of food image recognition on four subsets on ETH Food-101 dataset using the approach MobileNetV1 architecture.*

Methods	**No. of Parameters**	**Testing Time**
MobileNetV1	4.2 M	26m:40s
Modified MobileNetV1	3.2 M	24m:40s

Table 2 *The performance results of food image recognition on four subsets on ETH Food-101 dataset using the approach MobileNetV1 architecture.*

Methods	**Subsets of the ETH Food-101 dataset**			
	I	II	III	IV
Without data augmentation	45.84	51.29	60.26	66.78
Random cropping	45.79	55.82	59.52	67.44
With data augmentation	48.71	56.71	62.49	69.86
With data augmentation + random cropping	51.39	59.68	65.97	**72.59**

I show the obtained results of second to fourth experiments using the proposed MobileNetV1 architecture on four subsets of the ETH Food-101 dataset in Table 2. The table shows that the combination of the data augmentation and random cropping was the best approach in our experiments. This approach outperformed other methods with an increase of around 3-5% accuracy.

Table 3 *Performances of the five different techniques on ETH Food-101 dataset*

Methods	**The number of image per class**	**Accuracy**
Random Forest Discriminative Components (Bossard & Gool, 2014)	1000	50.76
Supervised Extreme Learning Committee (Martinel et al., 2016)	1000	55.89
Data Augmentation + MobileNet	400	57.90
Data Augmentation + Inception V3 (Yanai & Kawano, 2015)	1000	70.41
FoodNet: Ensemble Net (Pandey et al., 2017)	1000	72.10
DeepFood (C. Liu et al., 2018)	1000	77.00
Our proposed (Data Augmentation + MobileNetV1)	400	72.59
	1000	78.23

From the results in Table 3, the DeepFood architecture obtains the best performances on the ETH Food-101 dataset with an accuracy rate of 77%. Due to the computer used in the experiments, I decided to use the food image only 400 images per class to examine our proposed architecture. However, our modified MobileNetV1 architecture reached an accuracy of 72.59%. It is only 4.41% less than DeepFood architecture. As a result, our modified MobileNetV1 architecture outperforms the random forest discriminative components (Bossard & Gool, 2014), supervised extreme learning committee (Martinel et al., 2016) and three deep CNN architectures; MobileNetV1, Inception V3 (Yanai & Kawano, 2015) and FoodNet (Pandey et al., 2017). In addition, the modified MobileNetV1 created a model size of 22.4MB, which less than the MobileNet architecture 10MB.

2.6 Conclusion

In this study, I used the state-of-the-art MobileNetV1 architecture on the food image dataset. I also described a MobileNetV1 architecture, which was designed to address the overfitting problem. In this modified MobileNetV1 architecture, the number of parameters is decreased by applying the global average pooling (GAP)

layers. Moreover, the batch normalization (BN), rectified linear unit (ReLU), and dropout layers are combined. Also, the last layer is the softmax. In addition, the data augmentation techniques are computed before transferring to the training process.

From the experimental results, to the best of our knowledge, I trained the MobileNetV1 architecture according to the fine-tuned model. The modified MobileNetV1 architecture is competitive when compared to the original MobileNetV1 architecture on the ETH Food-101 dataset. I also demonstrated the impact of the data augmentation techniques; rotation, shift, flip, shear, zoom, and crop when implemented before assigning to the modified MobileNetV1 architecture to process. The best performance achieved when the combination of the various data augmentation techniques and the modified MobileNetV1 architecture.

In future work, I plan to construct the deep ensemble convolutional neural network (CNN) architectures, which are a combination of the state-of-the-art deep CNN architectures. I am interested in extracting the feature vector from the convolutional layers which may work better than individual deep CNN architecture.

Chapter 3
Deep Feature Extraction Techniques

There is a global increase in health awareness. The awareness of changing eating habits and choosing foods wisely are key factors that make for a healthy life. In order to design a food image recognition system, many food images were captured from a mobile device but sometimes include non-food objects such as people, cutlery, and even food decoration styles, called noise food images. These issues decreased the performance of the system. Convolutional neural network (CNN) architectures are proposed to address this issue and obtain good performance. In this chapter, I proposed to use the ResNet50-LSTM network to improve the efficiency of the food image recognition system. The state-of-the-art ResNet architecture was invented to extract the robust features from food images and was employed as the input data for the Conv1D combined with a long short-term memory (LSTM) network called Conv1D-LSTM. Then, the output of the LSTM was assigned to the global average pooling layer before passing to the softmax function to create a probability distribution. While training the CNN model, mixed data augmentation techniques were applied and increased by 0.6%. The results showed that the ResNet50+Conv1D-LSTM network outperformed the previous works on the Food-101 dataset. The best performance of the ResNet50+Conv1D-LSTM network achieved an accuracy of 90.87%.

3.1 Introduction

Overweight and obesity are the most significant factors for chronic diseases such as diabetes and cardiovascular diseases. The easiest way to avoid chronic diseases is to monitor and control people's dietary behavior. The advancement of artificial intelligence might help people to monitor and estimate daily calorie intake. Hence, food recognition systems are the most straightforward solution. Many systems can recognize several foods based on images. However, when people take a photograph several food characteristics (e.g. the shape and decoration of food, brightness adjustment, and non-food objects, called noise food images) are sent to the

system to compute and predict the food type and calorific content. These issues can be a cause of weaknesses of food imaging systems. Computer vision and machine learning techniques are proposed to address the problems mentioned above. Many researchers employ computer vision techniques to generate hand-crafted visual features and send robust features to the novel machine learning techniques, such as support vector machine (SVM), multilayer perceptron (MLP), random forest, and Naive Bayes techniques (Farooq & Sazonov, 2017; McAllister, Zheng, Bond, & Moorhead, 2018; Ragusa et al., 2016) to classify objects (Anthimopoulos, Gianola, Scarnato, Diem, & Mougiakkou, 2014; Martinel et al., 2016).

Furthermore, many studies have extracted the robust features, called the spatial features, using convolution neural network (CNN) architectures. The greatest benefit of this technique is that I can extract robust features with various CNN architectures. The robust features, however, are sent to be classified using traditional machine learning techniques. Additionally, the CNN architecture combined with a long short-term memory (LSTM) network has been applied for classification tasks. Nevertheless, a few researchers have invented CNN architectures and LSTM networks for food image recognition. In this research, I focus on improving the accuracy performance of the food image recognition based on CNN architectures and LSTM networks.

The significant contributions of this research are summarized in the following:

1. I propose the deep learning framework that combines state-of-the-art ResNet50, which is the convolutional neural network (CNN) and long short-term memory (LSTM) network, called ResNet50+Conv1D-LSTM network. This framework can extract robust features that are spatial and temporal features, from the food images. Mixed data augmentation techniques are also involved while training the CNN model. The data augmentation technique insignificantly increases the performance of food image recognition.

2. In these experiments, LSTM and Conv1D-LSTM networks were proposed to create robust temporal features. For the Conv1D network, various layers were combined, including zero padding, batch normalization, Convolution 1D, ReLU, batch normalization, dropout, and average pooling layers. In the training scheme, batch size, which was the number of training examples, were applied as 16, 32, and

64. The LSTM network results showed that a batch size of 32 provided a better result than batch sizes of 16 and 64.

Outline of the chapter. This chapter is organized as follows. Section 3.2 briefly explains deep learning researches in food image recognition systems and describes the different deep learning techniques. Section 3.3 describes the proposed approach for the food image recognition system. In section 3.4, the experimental settings and the results of the deep learning methods are presented. The conclusion and directions for future work are given in Section 3.5.

3.2 Related Work

In previous studies, many researchers have proposed using feature extraction methods based on handcrafted methods to extract features from images. Novel feature extraction methods such as local binary patterns (LBP) (Ojala et al., 1994), the scale-invariant feature transform (SIFT) (Lowe, 2004) the histogram of oriented gradients (HOG) (Dalal & Triggs, 2005), the speed-up robust features (SURF) (Bay et al., 2008) and a bag of visual words (BoVW) (Coates et al., 2011; Csurka, 2004) methods became popular and were proposed in many applications. Also, they achieved high accuracy performance. Secondly, the robust features extracted from the novel methods, are then given to machine learning algorithms such as support vector machine (SVM) (Cortes & Vapnik, 1995), K-nearest neighbor (KNN) (Altman, 1992), and multi-layer perceptron (MLP) for a task of classification.

The food image recognition, Anthimopoulos et al. (2014) proposed an automatic food recognition system to recognize 11 different central European foods. In the food recognition system, the features, namely visual words, are computed from the bag-of-features method and the k-means clustering algorithm. Then the linear SVM is used as a classifier. This method obtained a recognition performance of 78%. Furthermore, Martinel et al. (2016) introduced an extreme learning committee approach. This approach was divided into three parts; feature extraction methods, extreme learning committee, and supervised classification. First, various feature extraction methods were proposed to extract color, shape, texture, local, and data-driven features. Second, each feature vector was given to the extreme learning machine (ELM). Finally, the output from each ELM was sent to the SVM algorithm

for classification. The extreme learning committee outperformed the state-of-the-art methods on four benchmark food image datasets.

Deep learning techniques are becoming increasingly popular in food image recognition. In this section, I describe the research that has applied deep learning to solve the image recognition problem, including 1) deep learning for food image recognition and 2) deep feature extraction methods.

3.2.1 Deep learning for food images recognition

Convolution Neural Networks (CNNs) have been extensively used in food image recognition research. In 2016, Hassannejad et al. (2016) and Liu et al. (2016) used Google's image recognition architecture Inception. Hassannejad et al. (2016) proposed a network composed of 54 layers with fine-tuned architecture for classifying food images from three benchmark food image datasets: ETH Food-101, UECFOOD100, and UEC-FOOD256. On these datasets, the achieved accuracy was 88.28%, 81.45%, and 76.17%, respectively. Liu et al. (2016) invented the DeepFood network that modified the Inception module by introducing a 1×1 convolutional layer to reduce the input dimension to the next layers. It allows a less complicated network. The accuracy achieved was 77.40% with the ETH Food-101 dataset, 76.30%, and 54.70% with UEC-FOOD100, and UEC-FOOD256, respectively. In addition, the Inception architecture, the ResNet architecture is widely popular for food image recognition. Pandey et al. (2017) used ResNet, AlexNet, and GoogLeNet to propose an ensemble network architecture. The network consisted of three fine-tuned CNN in the first layer. All of the output was concatenated before being fed into ReLU nonlinear activation and passed to a fully connected layer followed by a softmax layer for image classification. Aguilar, Bolaños, and Radeva (2017) proposed the CNN Fusion methodology, which is composed of two main steps. First, training with state-of-the-art CNN models consisting of ResNet and Inception. Second, fusing the CNN outputs using the decision template scheme for classifiers fusion. The two proposed methods achieved accuracies of 72.12% and 86.71% with the ETH Food-101 dataset, respectively. Table 4 summarizes different food classification approaches. The accuracies reported along with the food databases used in the evaluation and the underlying CNN architecture

Table 4 *Performance evaluation of classification results on the food datasets using deep learning techniques.*

Datasets	**Architectures**	**Accuracy**	**References**
UEC-FOOD100 (Matsuda & Yanai, 2012)	DeepFood	76.30	Liu et al. (2016)
	InceptionV3	81.45	Hassannejad et al. (2016)
	WISeR	89.58	Martinel et al. (2018)
UEC-FOOD256 (Kawano & Yanai, 2015)	DeepFood	54.70	Liu et al. (2016)
	GoogLeNet	63.16	Bolanos and Radeva (2016)
	InceptionV3	76.17	Hassannejad et al. (2016)
	WISeR	83.15	Martinel et al. (2018)
ETH Food-101 (Bossard & Gool, 2014)	Inception	77.40	Lie et al. (2016)
	GoogLeNet	79.20	Bolanos and Radeva (2016)
	InceptionV3	88.28	Hassannejad et al. (2016)
	Ensemble Net	72.12	Pandey et al. (2017)
	CNNs Fusion	86.71	Aguilar et al. (2017)
	ResNet152	64.98	McAllister et al. (2018)
	WISeR	90.27	Martinel et al. (2018)

3.2.2 Deep feature extraction methods

Many researchers have focused on extracting features using several CNN architectures, called deep feature extraction (Y. Chen et al., 2016; Paul et al., 2016) that have been applied in many image recognition systems. With the deep feature extraction method, the pre-trained models of the state-ofthe-art CNN architectures are employed to train a set of images. Then, the deep features are extracted from the layer before the fully connected layer. After that, I can use the deep features as the input vector to a traditional machine learning algorithm, such as SVM, KNN, and MLP. Indeed, the state-of-the-art CNN architectures, such as VGG, ResNet, and Inception, have been proposed and widely used in the food image recognition system (Hassannejad et al., 2016; McAllister et al., 2018).

Table 5 *Performance evaluation of classification results on the food datasets using deep feature and machine learning techniques*

Datasets	Classes	Deep Feature Methods	Classifiers	Accuracy	References
PFID	7	AlexNet	SVM-linear	94.01	Farooq et al. (2017)
PFID	61	AlexNet	SVM-linear	70.13	Farooq et al. (2017)
UNICT-FD889	2	AlexNet	SVM-sigmoid	94.86	Ragusa et al. (2016)
Food-5K	2	ResNet152	SVM-RBF	99.4	McAllister et al. (2018)
Food11	11	ResNet152	ANN	91.34	McAllister et al. (2018)
RawFooT-DB	46	ResNet152	ANN	99.28	McAllister et al. (2018)
ETH Food-101	101	ResNet152	SVM-RBF	64.68	McAllister et al. (2018)

To classify the food and non-food images, Ragusa et al. (2016) proposed to use three deep feature methods called the Network in Network, the AlexNet, and the VGG-s models to extract features and then use a support vector machine (SVM) as a classifier. The best performance result was the AlexNet model combined with a binary SVM classifier on the Food-5k dataset. For multi-class food images, Farooq and Sazonov (2017) proposed the deep feature method called AlexNet to extract features from the PFID food image dataset. This method extracts the feature of 4,096, 4,096, and 1,000 channels from three fully connected (FC) layers; FC6, FC7, and FC8. Also, the linear SVM technique is applied as a classifier. The results showed that the features extracted from FC6 outperformed features from other FC layers. Moreover, McAllister et al. (2018) applied ResNet152 and GoogLeNet for deep feature methods performed on five datasets consisting of Food-5k, Food11, RawFooT-DB, and ETH Food-101 dataset. The deep features were then classified using traditional machine learning comprising SVM, artificial neural networks, Random Forest, and Naive Bayes. The experimental result with these methods had

accuracies above 90% on food image datasets, except for the ETH Food-101 dataset that obtained only 64.68% accuracy. A summary of food classification using the deep feature methods is shown in Table 5.

3.3 Proposed Approach for The Food Image Recognition System

This section explains the framework of food image recognition. Two main architectures, convolutional neural network (CNN) and long short-term memory (LSTM) network, are proposed to extract the robust features from the food images. Hence, the robust spatial and temporal features are extracted from state-ofthe-art ResNet architecture and the LSTM network. The temporal features extracted from the LSTM network are transformed into a probability distribution using the softmax function.

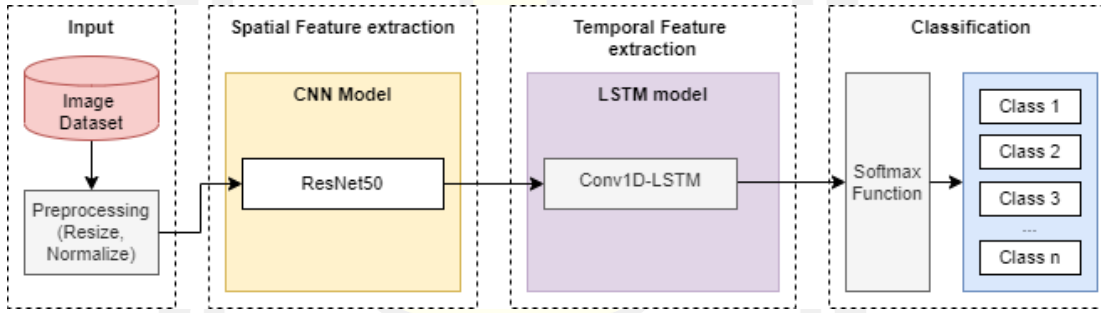


Figure 12 *Architecture of our proposed framework for food image classification.*

According to our framework, as shown in Figure 12, I examine the transfer learning strategy to train the ResNet architecture. Hence, this architecture considers only the color image and the resolution of the images is decided to be 224x224x3 pixels. I also normalize all food images to the values between 0 and 1 by dividing the pixel values with 255, which is the maximum value of the RGB color. Other schemes are described in the section of the spatial feature extraction method using CNN architecture and temporal feature extraction method using LSTM network, as follows.

In this section, I propose an effective CNN architecture to extract a robust spatial feature. According to the computation power and time, the transfer learning approach is applied in the training scheme, then the pre-trained models of CNN architectures are trained on the food image and then examined to discover the best robust spatial feature. As a result, the last pooling layer of the CNN model is

employed as the spatial feature, as shown in Figure 13. I can also call this method a deep feature extraction technique.

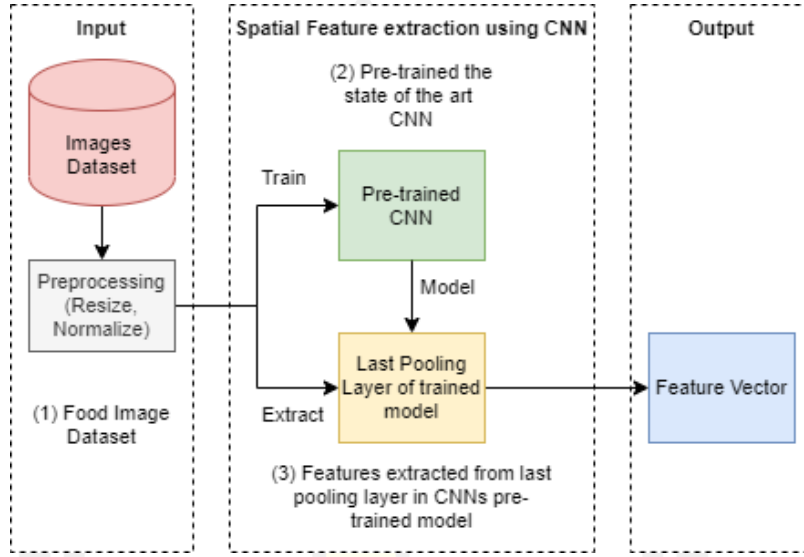


Figure 13 *Diagram of the deep feature extraction technique. (1) food images are fed to the pre-processing step to resize and normalize. In the spatial feature extraction process, (2) food images are trained using state-of-the-art CNN architectures to find weights with low validation loss. Then, (3) the spatial features of the food images are extracted according to the best CNN model.*

To extract the robust spatial features, in this study, I propose state-of-the-art CNNs, VGG16, VGG19, ResNet50, DenseNet201, MobileNetV1, and MobileNetV2. An overview of each CNN will now be described.

3.3.1 Spatial Feature Extraction using Convolutional Neural Network Architecture

3.3.1.1 VGGNet Architecture

Simonyan and Zisserman (2014) proposed a network to increase the stack of convolutional networks into 16 and 19 weight layers by using an architecture with a size of 3x3 pixels convolution filters, called VGGNet. With this network, the input images are the color image and are resized to 224x224 pixels resolution. The convolutional layers are downsized from 224x224 pixels to 7x7 pixels. Nevertheless, the number of feature maps is increased from 64 to 512 layers. The rectified linear unit (ReLU) is used as the activation function. Also, spatial pooling is computed by the max-pooling method with the size of a 2x2 pixel window.

Three fully connected (FC) layers follow VGGNet. The first two FC layers have 4,096 channels and the last FC layer contains 1,000 channels. The VGGNet is designed as a plain network, but still obtained the best performance on many image classification applications, such as remote sensing classification (X. Liu, Chi, Zhang, & Qin, 2018), and plant recognition (Abas, Ismail, Yassin, & Taib, 2018; Habiba, Islam, & Ahsan, 2019; Pearline, Vajravelu, & Harini, 2019).

3.3.1.2 ResNet Architecture

According to the plain network, the deeper convolutional layers were performed from 34-Layer until 152-layer plain networks (K. He, Zhang, Ren, & J., 2016). Firstly, the color image is resized to 224x224 pixels resolution and employed as the input of the deeper network. Secondly, the convolutional layers are divided into five convolutional blocks, namely building blocks. Remarkably, the output of each building block is always decreased by half of the input. For example, the output of the first, second, and fifth building blocks are 112x112, 56x56, and 7x7 pixels resolution, respectively. Finally, the average-pooling method is applied to the last building block and followed by the FC layer with 1,000 channels and the softmax function. As a result, the deeper plain network gave a higher error rate on the CIFAR-10 dataset.

(a) (b)

Figure 14 *Illustration (a) a building block and the residual function and (b) a sample of bottleneck network for ResNet 50, 101, and 152.*

According to the higher error rate, He et al. (2016) proposed to add the residual network, which is the shortcut connection, to train the deeper network, called ResNet. Hence, the shortcut connections are computed using the residual function that allows the network to skip two convolutional layers, as shown

in Figure 14a). The residual function is calculated by $F(x) = H(x) - x$ when the feature maps of the input and output have identical dimensions. The original function changes to $F(x) + x$. Furthermore, bottleneck architectures are presented when the deeper convolutional layers are implemented as 50, 101, and 152 layers. The bottleneck architectures allow the network to skip three convolutional layers, as shown in Figure 14b). Consequently, ResNet obtained a top-5 error rate of 3.57% on the ImageNet validation set and showed fast computation compared to the plain network. The ResNet also won the ILSVRC-2015 classification task.

3.3.1.3 DenseNet Architecture

Huang et al. (2017) proposed a dense network called DenseNet architecture. The different depth convolutional layers were experimented with consisted of 121, 169, 201, and 264. The result showed that the DenseNet with 264-layer provided the lowest top 1 error rate on the ImageNet validation set and yielded a better error rate than the ResNet architecture. Also, the parameter of the DenseNet is approximately 3-time less than the ResNet. According to the connection of the DenseNet, the network can connect to other layers in a feed-forward method. The number of direct connections can be computed using L(L+1)/2, where L is the number of layers. To further improve the DenseNet architecture, the convolutional layers are divided into four blocks, namely dense blocks. In each dense block, the bottleneck layers with a size of 1x1 and 3x3 convolution are used to reduce the number of input feature maps. The transition layers are combined with the dense blocks 1-3 to reduce the size of the feature maps to the half size of the convolutional layer in the dense block. The output size of each block is decreased from 112x112 to 7x7 pixels. As for the classification layer, the global average-pooling, FC layer, and softmax are applied. The differences between ResNet and DenseNet architectures are shown in Figure 15.

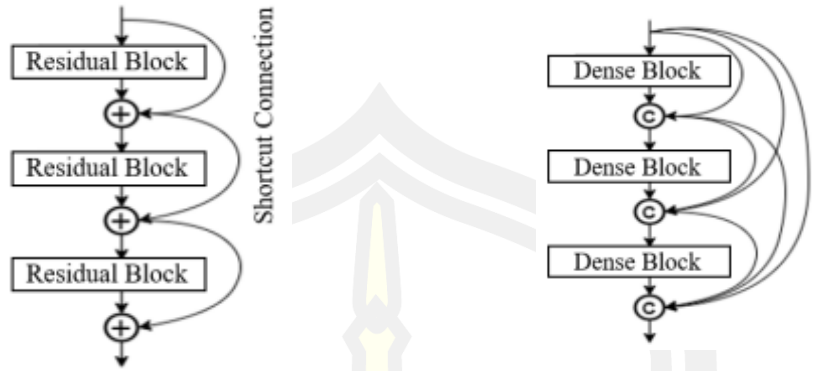


Figure 15 *Illustration of the difference of the connections between (a) the ResNet and (b) the DenseNet architectures.*

3.3.1.4 MobileNet Architecture

The lightweight CNN architecture called MobileNet is proposed for mobile and embedded devices (Howard et al., 2017). In order to reduce the size of the model, the depthwise separable convolution layer, a core layer of the MobileNet, is designed to factorize the standard convolution into 3x3 depthwise convolutions and then factorize the depthwise convolution layer into 1x1, called pointwise convolution. Due to MobileNet architecture, the depthwise and pointwise convolution layers are always followed by batch normalization (batchnorm) and ReLU, as shown in Figure 16a).

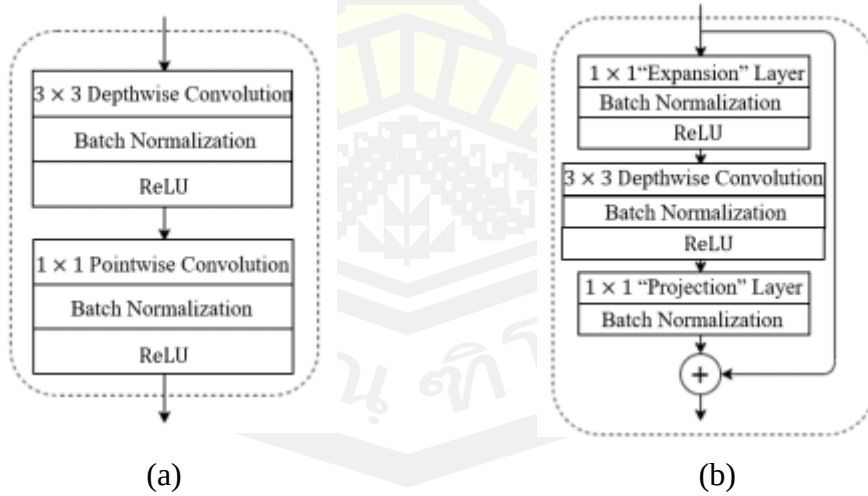


Figure 16 *Network architectures of MobileNet. Examples of (a) the depthwise separable convolution and (b) inverted residual and linear bottleneck.*

Furthermore, Sandler et al. (2018) proposed MobileNetV2 architecture. The new mobile architecture, called inverted residuals and linear bottlenecks, is combined with the linear bottleneck layer and inverted residual network. The inverted residuals and linear bottlenecks block consist of three layers. First, 1x1 convolution combined with batchnorm and ReLU. Second, depthwise convolution combined with batchnorm and ReLU. Third, 1x1 convolution combined with batchnorm and without non-linearity, as shown in Figure 16b). In MobileNetV2 architecture, the number of operations is decreased, so that is was of small size and low memory usage. A summary of the state-of-the-art CNN architectures is presented in Table 6.

Table 6 *Summary of the state-of-the-art CNN architectures.*

CNN Architectures	**Parameters**					
	No. of Conv Layer	**Filter Size**	**Stride**	**Pooling**	**No. of FC Layers**	**No. of Parameters**
VGG16	13	3	1	Max	3	138M
VGG19	16	3	1	Max,	3	143M
ResNet50	49	1, 3, 7	1, 2	Max, Average	1	25.6M
DenseNet201	200	1, 3, 7	1, 2	Max, Average	1	20.2M
MobileNetV1	13	1, 3	1, 2	Average	2	4.2M
MobileNetV2	13	3	1, 2	Average	1	3.2M

3.3.2 Temporal Feature Extraction

In this section, I propose two deep learning networks to extract temporal features, called long short-term memory and Conv1D-LSTM networks. The detail of deep learning networks is will now be described

3.3.2.1 Long Short-Term Memory

Hochreiter and Schmidhuber (1997) invented a novel gradient-based method and developed the network based on a recurrent neural network (RNN) called a long short-term memory (LSTM) network, as shown in Figure 18. It proposed to address the computational complexity, error flow, constraints of the feedforward neural network, and sequence problems of time series data (Jain, Gupta, & Moghe, 2018; Yan, Qi, & Rao, 2018). The LSTM network comprised special units that

connect to other units and are designed to cope with the sequence of data; video and speech data, called memory blocks. Each memory block contained the various functions consisting of the forget gate, input gate, update cell state, and the output gate.

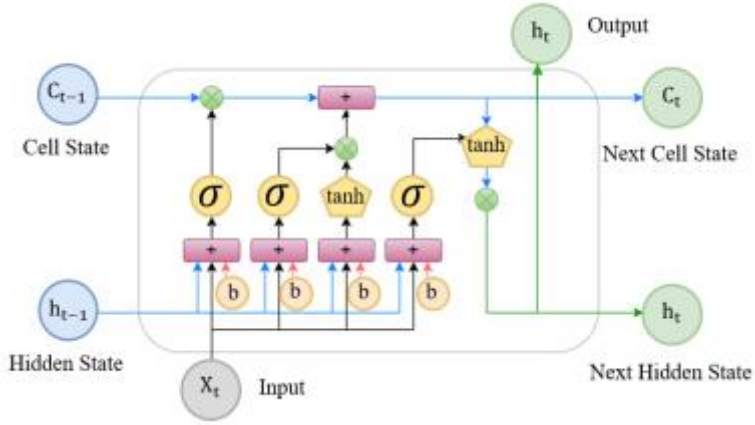


Figure 17 *The architecture of the long short-term memory network (Hochreiter & Schmidhuber, 1997).*

The memory block presented in Figure 17 is calculated as follows;

$$f_t = \sigma\left(w_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (1)$$

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\check{C}_t = tanh(w_c \cdot [h_{t-1}, x_t] + b_c)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \check{C}_t$$

$$O_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_f \cdot tanh(C_t)$$

where f_t is forget gate's activation vector, i_t is input/update gate's activation vector, $\check{C}_t$ is cell input activation vector, C_t is current cell memory, O_t is output gate's activation vector, h_t is current cell output, , b and W denote the bias vector and weight matrices for the input gate (i), output gate (o), forget gate (f), and memory cell (c), h_{t-1} is previous cell output, C_{t-1} is previous cell memory, σ is sigmoid function, and ' · ' is the Hadamard product (Hochreiter & Schmidhuber, 1997)

3.3.2.2 Conv1D-LSTM

In this study, I propose the Conv1D-LSTM framework to extract temporal feature from the spatial features, as shown in Figure 18. In the Conv1D block, the batch normalization layer was added so as to normalize the input data and speed up the process of learning. The dropout layer was implemented to prevent over-fitting, then some units were ignored during learning. After that, the average pooling layer which selected the average component from the sub-region of the feature map, was considered as the feature vector. The feature vector was sent to the LSTM Cells to learn and generate the temporal feature. Consequently, I again decreased the size of the feature using global average pooling layer (GAP) before giving the feature to the softmax function.

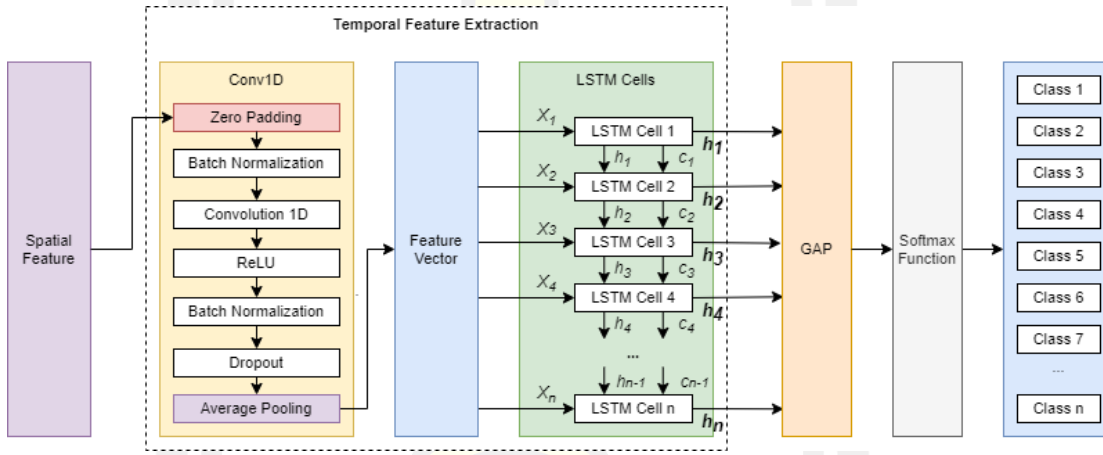


Figure 18 *Illustration of extract temporal features using the Conv1D-LSTM network.*

3.4 Experimental Setup and Results

3.4.1 Food Image Dataset

In this research, I focused on experimenting with the benchmark food image dataset, namely the ETH Food-101 dataset (Bossard & Gool, 2014). The training set contained the wrong labels and some noise images, such as food images taken from different camera angles that made other objects such as people, tables, and bottles, appear in the image. It consists of 75,750 training images and 25,250 test images. The sample images of the ETH Food-101 dataset are shown in Figure 19. The challenge of this dataset is that the training set contained some noise images, such as food images taken from different camera angles that made other objects such as

people, tables, and bottles, appear in the image, as shown in Figure 20(a) and similarities of shape, color, and decoration between two categories (chocolate cake and chocolate mousse), as shown in Figure 20(b). The researchers assume that computer vision can handle noise images and wrong labels.

Figure 19 *Sample images of the ETH Food-101 dataset*

Figure 20 *Some examples of the ETH Food-101 dataset that containing (a) other objects (e.g., people, cake shelves, tables, and glasses of beer) and (b) similarities of chocolate cake and mousse.*

3.4.2 Experimental Setup

As explained in Section 3, I first used pre-trained models of six CNN architectures; VGG16, VGG19, ResNet50, DenseNet201, MobileNetV1, and MobileNetV2, to train and extract the spatial feature from food images. All CNNs were trained using the stochastic gradient descent (SGD) optimizer, rectified linear unit (ReLU) for activation function, and learning rate between 0.01 to 0.0001. Second, the spatial features were then sent to Conv1D-LSTM and LSTM networks to

extract temporal features. In the LSTM network, the fraction of the units was employed to drop the linear transformation of the inputs. The initial weights were randomly selected by using a Gaussian distribution where the mean is zero.

I decided to train only 100 epochs to avoid overfitting when training the model. Figure 21 shows loss values while training the Conv1D-LSTM and LSTM model. According to loss values, better loss values were obtained after epoch 50 when they became stable values until epoch 100.

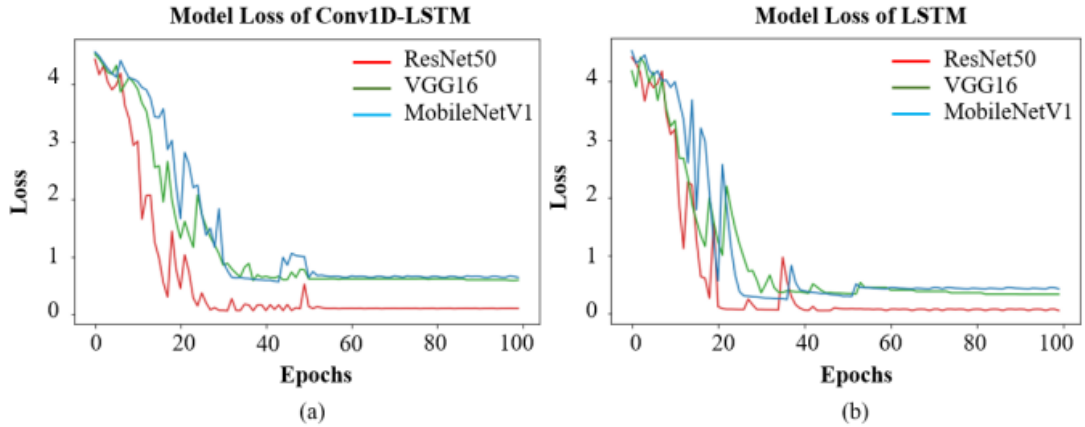


Figure 21 *Illustration of loss values of (a) Conv1D-LSTM and (b) LSTM networks when using ResNet50, VGG16, and MobileNetV1 as a deep feature method.*

3.4.3 Evaluation Metrics

The evaluation metrics used for food image recognition were accuracy and F1-score. I used the accuracy score to evaluate the performance of the deep learning models on the test set and used the F1-score to examine the individual accuracy of each class. The accuracy and the F1-score were computed by Equations 2 and 3.

$$accuracy = \frac{TP_k+TN_k}{TP_k+TN_k+FP_k+FN_k} \quad (2)$$

$$F1-score = 2 \times \frac{\left(\frac{TP_k}{TP_k+FN_k} \times \frac{TP_k}{TP_k+FP_k}\right)}{\left(\frac{TP_k}{TP_k+FN_k} + \frac{TP_k}{TP_k+FP_k}\right)} \quad (3)$$

where TP_k called true positives, is the number of correctly classified images from class k, FP_k called false positives, is the number of misclassified images from class k. TN_k called true negatives, is the number of correctly classified image that does not

belong to class k, and FN_k called false negatives is the number of misclassified images belong to class k.

3.4.4 Experiments with Deep Learning Methods

In the experiments with deep learning methods, I first trained the ETH Food-101 dataset using a pre-trained model of six state-of-the-art CNNs; VGG16, VGG19, MobileNetV1, MobileNetV2, ResNet50, and DenseNet201. Second, I proposed the deep feature method to extract the spatial feature from the last pooling layer of each CNN. The deep feature method extracted a high dimension of the spatial feature. The number of spatial features is reported in Table 7. It can be seen that ResNet50 provided 99,176 features. On the other hand, VGG16 produced only 25,088 features. Finally, I trained the high dimension of the spatial features using Conv1D-LSTM and LSTM networks.

Table 7 *Illustration of the number of spatial features extract from different CNN architectures and size of each model*

Deep Feature Methods	No. of Parameters	No. of Features
VGG16	14.7M	25,088
VGG19	20M	25,088
ResNet50	23.5M	99,176
DenseNet201	18.3M	94,080
MobileNetV1	3.2M	50,176
MobileNetV2	2.2M	62,720

Table 8 and Figure 22 present the accuracy results on the test set of the ETH Food-101 dataset for CNN, Conv1D-LSTM, and LSTM networks. The results show that the Conv1D-LSTM achieved the best performance with 89.82% accuracy when using a batch size of 32 and extracting features with ResNet50. As a result, the Conv1D-LSTM network with the batch size of 32 always showed better accuracy than other batch sizes. According to our experiments, however, the CNN architectures presented worse performance compared to the Conv1D-LSTM and LSTM networks. In terms of the deep feature methods, the ResNet50 outperforms all CNN architectures when training with the CNN, Conv1D-LSTM, and LSTM networks. The result of the CNN architectures shows that the ResNet50 provided 42.66% accuracy

higher than the MobileNetV2. I concluded that the ResNet50 extracted the spatial feature with a high dimension and still provided higher accuracy when training with Conv1D-LSTM and LSTM networks. Hence, the ResNet50 combined with the Conv1D-LSTM, namely ResNet50+Conv1D-LSTM, performed best on the ETH Food-101 dataset.

Table 8 *Evaluation of the classification results for the ETH Food-101 dataset using different deep learning consisting of CNN, LSTM, and Conv1D-LSTM. The first column shows the deep feature methods that used to extract spatial features.*

Model	**CNN**	**LSTM**		**Conv1D-LSTM**		
		No Pooling Layer	**Global Average Pooling**	**No Pooling Layer**	**Average Pooling**	**Max Pooling**
VGG16	67.40	78.55	80.44	75.94	85.91	84.61
VGG19	65.54	77.15	79.94	75.02	85.66	84.52
MobileNetV1	50.60	58.59	60.32	64.80	65.88	65.75
MobileNetV2	37.20	50.33	51.94	55.14	56.73	56.71
DenseNet201	39.29	38.08	38.98	42.25	42.87	38.11
ResNet50	79.86	88.90	88.92	86.83	**89.82**	89.01

The experimental results show that the Conv1D-LSTM outperformed LSTM because I combined necessary layers toward the Conv1D network, such as batch normalization, ReLU activation function, and dropout. These layers produced the Conv1D network to normalize the inputs to each feature map and cope with the linear activation function. For Conv1D, I experimented with pooling layers; global average pooling and global max pooling to decrease the size of the feature vector before giving it to classified with the softmax function. The success of the pooling layer is no parameter to optimize and robust to perform the spatial feature.

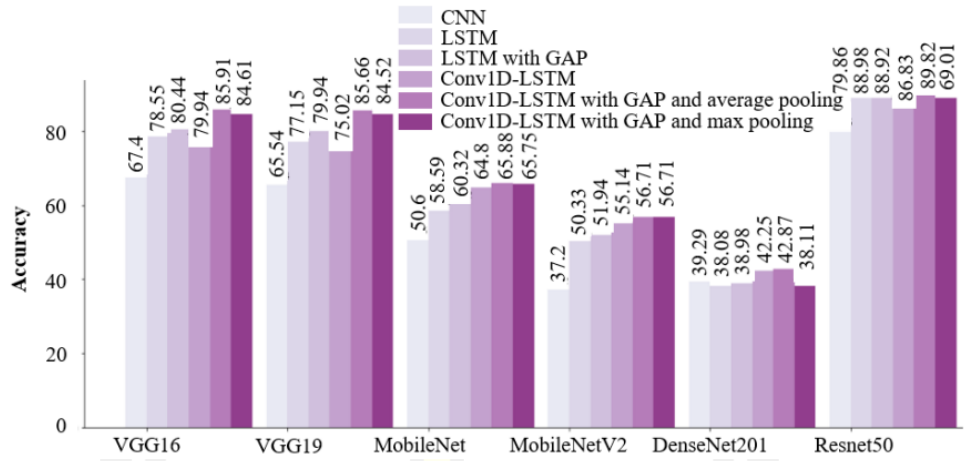


Figure 22 *Performance evaluation of three classifiers consisted of CNN, Conv1D-LSTM, and LSTM architectures that extract features based on six different deep CNN architectures on the ETH Food-101 dataset.*

To study the effect of the data augmentation techniques, I applied six data augmentation techniques; rotation, width shift, height shift, horizontal flip, shear, and zoom while training the CNN architecture because Phiphiphatphaisit and Surinta (2020) reported that data augmentation techniques could increase the accuracy of CNN, especially for food image recognition. In this experiment, ResNet50+Conv1D-LSTM using the batch size of 32 was considered.

Table 9 *The classification results for the ETH Food-101 dataset using features that extracting from the ResNet50 architecture and data augmentation techniques.*

Data Augmentation	LSTM	Conv1D-LSTM
No	88.92	89.82
Yes	89.49	90.87

Table 9 showed that LSTM and Conv1D-LSTM perform better when data augmentation techniques were applied. The accuracy of the Conv1D-LSTM with the data augmentation technique was slightly increasing compared with the LSTM with the data augmentation technique. As a result, the ResNet50+Conv1D-LSTM network with the data augmentation technique provided an accuracy of 90.87% on the ETH Food-101 dataset. The data augmentation can generate more food images while training, and then it increases the robustness of the model without decreasing the

effectiveness. Table 10 showed number of parameters and testing time of ResNet50 and ResNet50+Conv1D-LSTM.

Table 10 *The performance results of food image recognition on four subsets on ETH Food-101 dataset using the approach MobileNetV1 architecture.*

Methods	No. of Parameters	Testing Time
ResNet50	24.6 M	30m:50s
ResNet50+Conv1D-LSTM	38.3 M	32m:30s

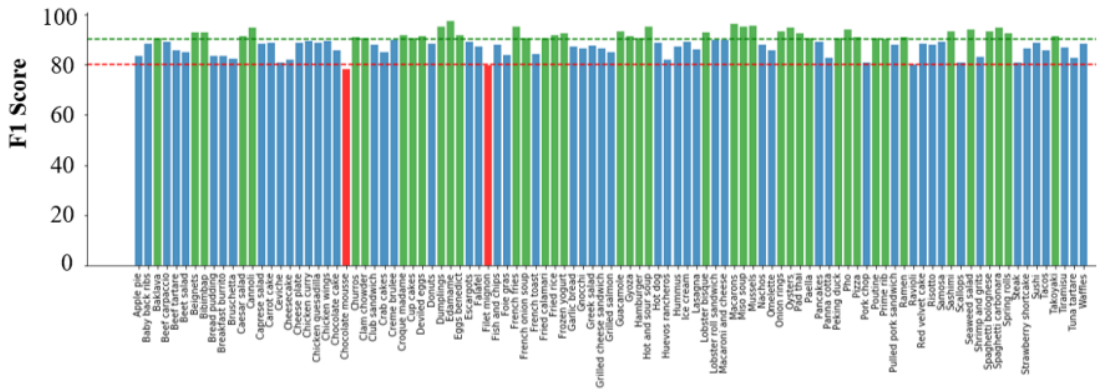


Figure 23 *The result of the F1-score on the ETH Food-101 dataset using the ResNet50 and LSTM architectures.*

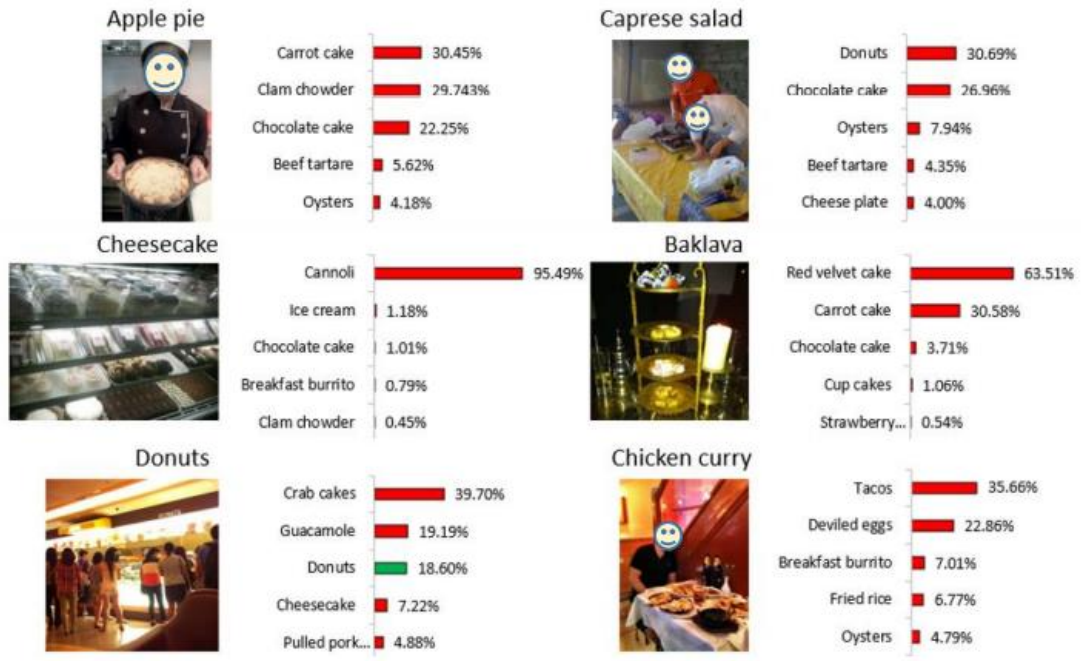


Figure 24 *Examples of misclassified results according to the noise images.*

The F1-score value of the ResNet50+Conv1D-LSTM network was computed according to Equation (3) and is illustrated in Figure 23. I found that only two categories, chocolate mousse and Filet mignon (see red bar) provided an F1-score of less than 80%. The F1-score also reported that 42 categories (see green bar) obtained a score above 90%. However, when I examined the ResNet50+Conv1D-LSTM network with non-food elements, called noise images, our proposed network could not classify these noise images correctly. Some noise images are shown in Figure 20a) and the misclassified results of the noise images are shown in Figure 24. Also, misclassification of similar categories such as chocolate cake and chocolate mousse were found, as shown in Figure 25.

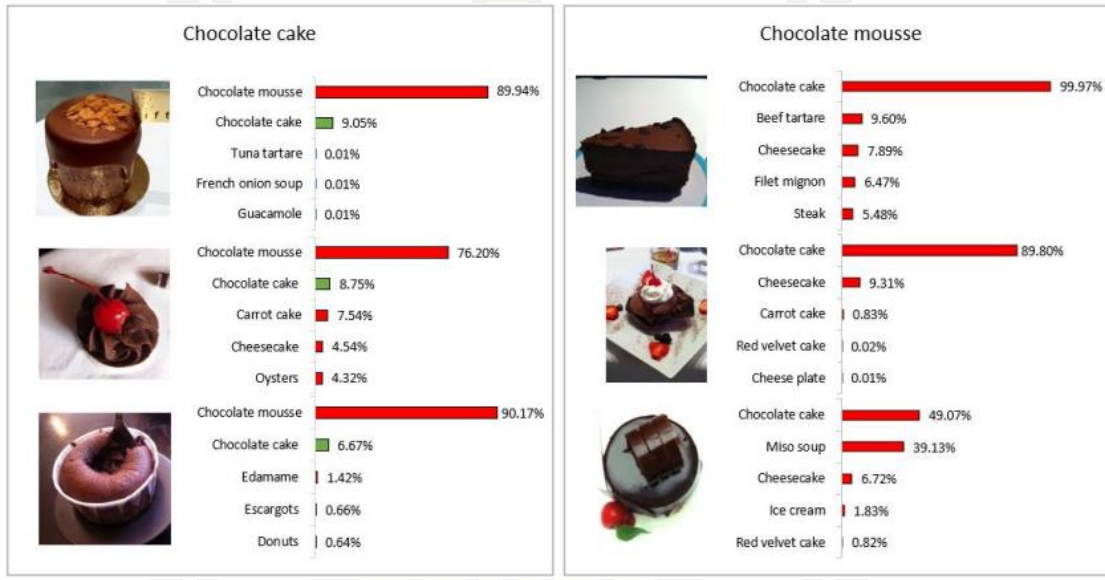


Figure 25 *An example of the similarity categories between chocolate cake and chocolate mousse contains in the ETH Food-101 dataset.*

4.4.5 Comparison between ResNet50+Conv1D-LSTM Network and Previous Methods

I made extensive comparisons between our ResNet50+Conv1D-LSTM network and existing state-of-the-art CNN architectures. The experimental results showed that our network performed better than all CNN architectures. The accuracy of 90.87% was obtained from the ResNet50+Conv1D-LSTM, while, the performance of the state-of-the-art WISeR architecture was 90.27% accuracy. The comparative

results between the existing CNN architectures and our proposed architecture on the ETH Food-101 dataset are shown in Table 11.

Table 11 *Recognition performance on the ETH Food-101 dataset when compared with different deep learning techniques.*

Architectures	No. of training images per class	Accuracy	References
ResNet152	750	64.98	McAllister et al. (2018)
EnsembleNet	750	72.12	Pandey et al. (2017)
Modified MobileNetV1	400	72.59	Phiphiphatphaisit & Surinta (2020)
DeepFood	750	77.40	Liu et al. (2016)
GoogLeNet	750	79.20	Bolanos & Radeva (2016)
CNNs Fusion	750	86.71	Aguilar et al. (2017)
InceptionV3	750	88.28	Hassannejad et al. (2016)
WISeR	750	90.27	Martinel et al. (2018)
ResNet50+Conv1D-LSTM	750	90.87	Our proposed

From the experimental results shown in Table 11, it can be seen that the Conv1D-LSTM yielded better performance than other techniques. Our Conv1D network included many layers consists of batch normalization layer, ReLU activation function, and dropout layer. In our Conv1D, I used the batch normalization layer to normalize the input data to each feature map and this layer works better with the ReLU activation function. The dropout layer was attached to the Conv1D network to prevent the over-fitting, then it allows the network to ignored some units during training.

3.5 Conclusions

This study proposed the ResNet50+Conv1D-LSTM network for accurate food image recognition. First, our network took advantage of extracting the robust spatial feature using a state-of-the-art convolutional neural network (CNN), called ResNet50 architecture. Second, I used the robust feature as input data for the Conv1D combined

with the long short-term memory (LSTM) network, namely Conv1D-LSTM. The primary function of the Conv1D-LSTM network was to extract a temporal feature. Finally, the softmax function was employed to transforms the output of the Conv1D-LSTM into a probability distribution.

In the experiments, I evaluated six CNNs; VGG16, VGG19, ResNet50, DenseNet201, MobileNetV1, and MobileNetV2 to extract the feature, then classify with Conv1D-LSTM and LSTM networks on the Food101 dataset. The results showed that the ResNet50 combined with the Conv1D-LSTM network, called ResNet+Conv1D-LSTM network, provided the best performance (see Table 7). Additionally, I experimented with mixed data augmentation techniques; rotation, width shift, height shift, horizontal flip, shear, and zoom. The result of the data augmentation also insignificantly increased accuracy by 0.27%. Our experiments presented better results than previous work (see Table 9). The best result of the ResNet+Conv1D-LSTM obtained 90.87% on the ETH Food-101 dataset.

In future work, I will experiment on increasing the performance of the food image recognition. I will consider other novel data augmentation techniques, which could be more efficient in the noise food images. Also, the ensemble and parallel networks will be involved in future work.

Chapter 4
Adaptive Deep Feature Learning Techniques

Various deep learning methods have been proposed to address the challenge of food image classification, such as convolutional neural networks (CNN), deep feature extraction, and ensemble CNNs. However, the existing methods do not perform with high accuracy on the benchmark food image datasets. In this research, I proposed a robust adaptive spatial-temporal feature fusion network, called ASTFF-Net, to enhance the performance of the food image recognition system. The architecture of ASTFF-Net is divided into three parts; spatial feature extraction network, temporal feature extraction network, and adaptive feature fusion network. In the first part, I extracted the spatial features using the ResNet50 and then minimized the size of the parameters using the reduction operation. Further, the convolutional 1D (Conv1D) block was applied to fit the features into the recurrent neural networks. In the second part, the spatial features from the first part were given to the long short-term memory (LSTM) that allows learning various patterns from sequence features. In the final part, the spatial features from the first part and temporal features from the second part were concatenated and assigned to the Conv1D, followed by the softmax layer. The advantage of ASTFF-Net is that the proposed network can prevent overfitting problems due to the attachment of the global average pooling and dropout layers. These layers decreased the number of network parameters and dropped the number of connections between layers, respectively. In the experiments, I evaluated four different adaptive feature fusion networks (ASTFF-NetB1 to B4) on four benchmark food image datasets; Food11, UEC Food-100, UEC Food-256, and ETH Food-101. As a result, the proposed ASTFF-NetB3 achieved the best performance on four benchmark food image datasets. It also significantly outperformed the existing methods.

4.1 Introduction

Nowadays, people care about their health and make sure they live a fit and good life. Many food image recognition applications, such as dietary, personal food logging, nutrition assessment, and social media applications (Jiang et al., 2020; C. Liu

et al., 2016; Sahoo et al., 2019; Dong, Sun, & Zhang, 2019; Nordin, Xin, & Aziz, 2019), were invented to yield the users' requirements. In order to use the program to its full potential, many applications were then built as mobile applications on smartphones. They allow people who use the smartphone to take food photos and measure nutrition themselves.

To make the food image recognition applications achieve more accurate results in classification, the artificial intelligence algorithms should deal with uncontrolled photos taken by the users with variations such as brightness, orientation, noise, and other objects in the food images. Figure 26a shows some different orientations of spaghetti. The Peking duck, as shown in Figure 26b, is decorated in different styles. Furthermore, Figure 26c shows other objects in the food images, such as glasses, plates, forks, spoons, and knives. Many techniques have been proposed to address these challenges.

(a) (b) (c)

Figure 26 *Illustrated food images (a) similarities in different food types (b) different decoration and (c) non-food items.*

Many convolutional neural network (CNN) architectures are currently proposed for food image recognition systems that make it more effective to analyze and classify real-world food images. CNNs have also shown state-of-the-art performance on food image recognition. The fine-tuned models of AlexNet and InceptionV3 architectures were used to recognize the real-world food images on the benchmark food image datasets; ETH Food-101, UEC Food-100, and UEC Food-256 (Yanai and Kawano, 2015; Hassannejad et al., 2016). In their experiments, Yanai and Kawano (2015) obtained the recognition accuracy of 78.77% and 65.57% on UEC Food-100 and Food-256, respectively. In comparison, Hassannejad et al. (2016) achieved an accuracy of 88.28% on ETH Food-101, 81.45% on UEC Food-100, and 76.17% on UEC Food-256.

The concept of the ensemble CNNs network, called Ensemble Net, was proposed by Pandey et al. (2017). In their ensemble Net, the input images were first changed to HSV color space and then histogram equalization was applied to only the brightness channel. Second, the food images were sent to fine-tuned CNNs consisting of AlexNet, GoogLeNet, and ResNet. Third, the feature maps that had been extracted from three CNNs were concatenated and sent to the fully connected layers. Finally, their proposed network was classified using the softmax function. Ensemble Net performed with a recognition accuracy of 72.12% on the ETH food-101 and 73.5% on the Indian food database.

The deep feature extraction technique became the popular method that extracted the robust deep features based on the convolutional neural networks (CNNs). The CNN architecture emphasizes that it computes the weighted parameters from the input images and then creates unique spatial features. Şengür et al. (2019) extracted deep features using two CNN architectures; VGG16 and AlexNet. The deep features were then concatenated and sent to classify using the support vector machine (SVM) technique. Phiphitphatphaisit and Surinta (2021) extracted both spatial and temporal features. First, the spatial features were extracted using ResNet50 and spatial features were subsequently transferred to the Conv1D-LSTM network to extract the temporal features. Finally, the deep features were classified using the softmax function.

To better extract the unique deep features from real-world food images, the significant contributions of this thesis are summarized in the following. I introduce a novel CNN-based network for encoding food images to extract robust deep features, namely adaptive spatial-temporal feature fusion network (ASTFF-Net). ASTFF-Net has three main networks; spatial feature extraction, temporal feature extraction, and adaptive feature fusion. The advantage of our proposed network is that it captures the spatial and temporal to represent real-world food image characteristics. I then show that ASTFF-Net significantly outperforms existing state-of-the-art deep learning techniques on four real-world food image datasets; Food11, UEC Food-100, UEC Food-256, and ETH Food-101.

The remainder of this chapter is organized as follows. Section 4.2 summarizes the overview of related work. Section 4.3 describes the proposed ASTFF-Net. The

real-world food image datasets are explained in Section 4.4. The experimental results and discussion are presented in Section 4.5. The conclusion and future work are given in Section 4.6.

4.2 Related Work

Recently, many approaches have been proposed to address the challenge of real-world food image recognition. The related works are described in this section, including convolutional neural networks, deep feature extraction methods, and deep feature fusion methods.

4.2.1 Convolutional Neural Networks (CNNs)

CNN architectures are popular and have been proposed to address the recognition problems in many domains. Many CNN architectures were proposed to recognize food images, such as VGG16, GoogLeNet, InceptionV3 (Hassannejad et al., 2016; Liu et al., 2016; Ege and Yanai, 2017; Vijayakumar and Sneha, 2021). Ng et al. (2019) proposed to use several state-of-the-art CNN architectures comprising MobileNetV2, ResNet50, InceptionV3, InceptionResNetV2, Xception, and NASNet-Large for food image recognition. In their experiments, they evaluated the performance of the CNN architectures on several parameters, including the impact of the training images, data augmentation techniques, class imbalance, and image resolutions. The results showed that the Xception perform better than other CNNs on UEC Food-100, ETH Food-101, and Vireo-Food 172 datasets.

Martinel et al. (2018) invented wide-slice residual networks (WISeR) based on a residual network. The WISeR architecture contained two parts; residual network and slice network. In the first part, the residual network was employed. In the second part, the slice convolution kernel was proposed. The slice convolution kernel was designed using the rectangle kernel. The width of the rectangle kernel was the same size as the width of the input image. It was different from the standard convolution kernel in that the kernel of the standard convolution was designed as the square kernel. Further, two parts were concatenated and given to fully connected layers. The WISeR architecture obtained an accuracy of 89.58% on the UEC Food-100, 83.15% on the UEC Food-256, and 90.27% on the ETH Food-101 datasets.

Moreover, Tasci (2020) proposed ensemble CNNs using voting combination rules, called voting-based CNNs. For the CNN architectures, five CNNs, including VGG16, VGG19, GoogLeNet, ResNet101, and InceptionV3, were experimented with. In the ensemble method, six voting methods (minimum, average, median, max, product, and weighted probabilities) were evaluated. The voting-based CNNs yielded 84.28%, 84.52%, and 77.20% accuracy rates on ETH Food-101, UEC Food-100, and UEC Food-256 respectively.

4.2.2 Deep Feature Extraction methods

Deep feature extraction methods aim to extract the spatial features from the input images. They are designed to extract features from different layers of deep CNN architectures to enhance accuracy performance. Hence, the deep features are transferred to the recurrent networks and other machine learning techniques to train and create a robust model. Further, the deep features can also be assigned to the LSTM network to extract the temporal features.

Ragusa et al. (2016) used AlexNet, VGG, and Network-in-Network models to extract the deep features from food images. The deep features were then given to classify using the support vector machine (SVM) techniques. The results showed that extracted deep features using AlexNet architecture and classified using the binary SVM outperformed extracted deep features using other CNNs. As a result, training the binary SVM technique on the deep features performed approximately 8% better than classification using only the CNN technique.

Aguilar et al. (2017a) proposed to use GoogLeNet architecture as the feature extraction method. In their method, first, the deep features were transformed and the best discriminant components selected using principal component analysis (PCA). Second, the best components were trained using the SVM technique. Moreover, in SVM, the grid-search method was used to find the best hyperparameters; cost and gamma. Finally, the optimal SVM model was trained on the best components with the best hyperparameters, then the input images were classified as the food or non-food images. It obtained an accuracy of 94.86% on the RagusaDS and 99.01% on the FCD datasets.

The idea of extracting the deep features from various convolution layers was proposed by Farooq and Sazonov (2017). In their method, the deep high-level features were extracted from convolution layers 6, 7, and 8 of the AlexNet architecture. It extracted 4,096, 4,096, and 1,000 features from the images, respectively. Consequently, the SVM classifier trained deep features from layers 6, 7, and 8 separately. As a result, the extracted deep feature from layer 6 achieved the highest accuracy with 70.13% on the Pittsburgh fast-food image dataset. Furthermore, McAllister et al. (2017) extracted the deep features using ResNet-152 and GoogLeNet architectures from food image datasets. The deep features were then classified using four classifiers consisting of SVM, random forest, neural network, and Naive Bayes. The experimental results showed that it obtained a very high accuracy of 99.4% on the Food-5k dataset. Subsequently, it obtained an accuracy above 90% on Food11 and RawFooT-DB datasets. However, it achieved only 64.98% on the ETH Food-101 dataset.

4.2.3 Deep Feature Fusion Methods

The previous research mentioned above has shown that deep CNN features achieve high performance in classifying food images. In this section, I will discuss deep feature fusion for food image recognition. Pandey et al. (2017) presented a fusion of three deep CNN features consisting of AlexNet, GoogLeNet, and ResNet to classify benchmark food datasets. In the first layer, three fine-tuned CNNs were used for feature extraction, and the output was concatenated before being passed to ReLU activation followed by a fully connected layer and fed into the softmax function for classification. The experimental result on the ETH Food-101 dataset achieved 72.12% accuracy. Aguilar et al. (2017b) proposed the CNN fusion method based on Inception Modules and Residual Networks. The first step involved separately training two CNN models. Second, the best results in the validation dataset were used in the fusion step using the decision template scheme. The method achieved an accuracy of 86.71% with the ETH Food-101 dataset.

In addition to the featured fusion methods, adaptive feature fusion has also been introduced for image classification. For example, Li et al. (2020) proposed multi-exemplar images and adaptive fusion of features to enhance blind face

restoration. Kumar, Namboodiri, and Jawahar (2020) used the adaptive feature aggregation to recognize a person. The method was to combine the pooled features from multiple locations of the shared feature maps with adaptive weights produced by the attention module. Zhao et al. (2021) introduced a tracking algorithm with a multi-level adaptive feature fusion method. From all the research mentioned above, it was found that the adaptive feature fusion approach increases the efficiency of image recognition. In our study, I used the deep feature technique to extract the feature of the food image and fused the feature with the adaptive spatial-temporal feature fusion method, descript as follows in section 4.3.

4.3 Adaptive Spatial-Temporal Feature Fusion Network (ASTFF-Net)

Overview of the network. The architecture of the adaptive spatial-temporal feature fusion network, called ASTFF-Net, is shown in Figure 27. ASTFF-Net is proposed to improve the robustness of the deep features extracted using the deep learning methods. It is divided into three schemes; spatial feature extraction network, temporal feature extraction network, and adaptive feature fusion network.

For the first scheme, the deep features are extracted using the deep convolutional neural network (CNN) from food images, with ResNet50 architecture. The reduction operation is then applied to minimize the size of the network parameters. Further, I provide the data to fit the recurrent neural networks, such as long short-term memory networks (LSTMs) and gated recurrent units (GRUs), by applying the convolutional 1D block. I describe the details of the spatial feature extraction network in Section 4.3.1.

For the second scheme, the spatial features from the previous scheme are assigned to the LSTM network to extract the temporal features. The LSTM network was proposed by combining feedback connections to learn many sequence tasks. The details of the LSTM network are explained in Section 4.3.2.

In the last scheme, I concatenate the spatial and temporal features to obtain the advantages from these features, called the adaptive feature fusion network. In addition, the convolutional 1D (Conv1D) block is attached to the temporal and spatial feature extraction networks and then combined using concatenate operation. The explanation of the adaptive feature fusion network is shown in Section 4.3.3.

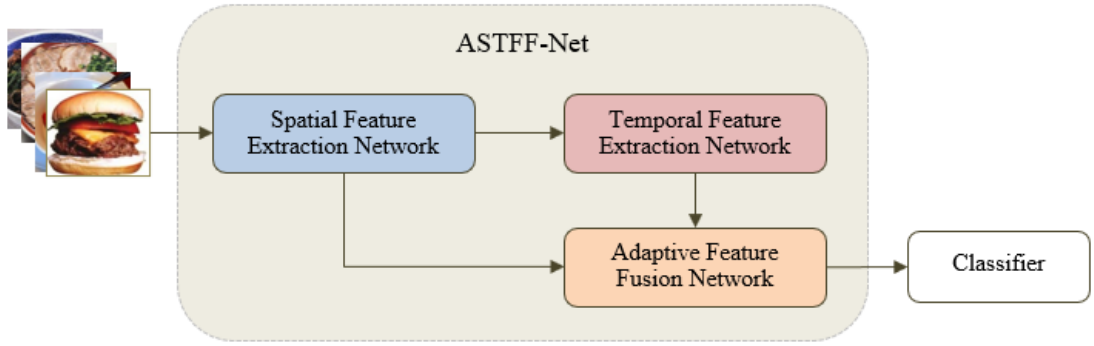


Figure 27 *Overall of our ASTFF-Net*

4.3.1 Convolutional Operations

This section briefly describes the convolutional operations involved in the experiments, including convolutional 1D, batch normalization, rectified linear unit, dropout, and pooling layers, as follows.

4.3.1.1 Convolutional 1D

The convolutional layer is the principal layer of CNN architecture (LeCun et al., 1998) proposed to extract the spatial feature from the 2D image. The convolution operation was used to calculate between the input image and the small square filter. The output of the operation is recognized as a feature map. The convolutional operation is calculated as follows.

$$x_j^l = \sum_{l=1}^{n} x_i^{l-1} \times w_{ij}^{l-1} + b_j^l \qquad (1)$$

where x_j^l is the j^{th}feature map in layer l, x_i^{l-1} is the i^{th}feature map in layer $l-1$, w_{ij}^{l-1} is weights of the j^{th} filter that can be updated while training the network, and b_j^l is the trainable bias parameter of the j^{th}feature map in layer l.

Furthermore, the convolutional layer was applied to deal with the 1D vector, called convolutional 1D (Conv1D). Therefore, Conv1D was applied to the natural language processing (NLP) and forecasting tasks. In our experiments, the filter size of 1x3 with a stride of 1 was applied to the Conv1D.

4.3.1.2 Batch Normalization

The batch normalization (BN) is proposed due to the parameters of the previous CNN layers changing during the training process of the

CNN model (Ioffe and Szegedy, 2015). It provides a uniform distribution before sending the weighted parameter to the further CNN layer. The BN benefits by being to defeat the vanishing gradient problem, because the slight variations in parameters from the current layer to the following layers do not get propagated. Consequently, it is possible to use higher learning rates to optimize the model and result in faster training.

The BN operation is computed as follows: 1) Calculate mean (μ_B) and variance (σ_B^2) of mini-batch (B) (see Equations 2 and 3), 2) Normalize input (x) by subtracting x with μ_β and dividing mini-batch standard deviation (see Equation 4), and 3) Scale and position the dataset by applying norm($\widehat{x_\iota}$) to calculate with scaling parameter (γ) and shifting parameter (β) (see Equation 5), which will be added to backward propagation to allow the algorithm to adjust both values during training the model.

$$\mu_\beta \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2}$$

$$\sigma_\beta^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m}\left(x_i - \mu_\beta\right)^2 \tag{3}$$

$$\widehat{x_\iota} \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \tag{4}$$

$$y_i \leftarrow \gamma\widehat{x_\iota} + \beta \equiv BN_{\gamma,\beta}(x_i) \tag{5}$$

where β is values of x over a mini-batch, $\epsilon$is the smoothing term that guarantees stability numeric within the operation by stopping a division by a zero value, and m is the input numbers in the mini-batch.

4.3.1.3 Rectified Linear Unit

Rectified Linear Unit (ReLU) is an activation function often applied in neural networks (Nair and Hinton, 2010) which has simple and not heavy computation. Hence, the CNN model could require less training time. The ReLU function is designed as a linear function that returns zero if it gets any negative input. Otherwise, the function returns the same value for any positive input value. The ReLU function is computed by $f(x) = max(0, x)$, where x is the input value.

4.3.1.4 Dropout

The robust network has many weighted layers. It usually contains a large number of parameters that have to be adjusted. The overfitting problem may occur while training the network. To address the overfitting problem, in this study, I proposed to apply the dropout layer (Srivastava et al., 2014) in our proposed network. With the dropout layer, the neural nodes and their connections were randomly dropped during training the model.

4.3.1.5 Pooling Layers

1. Average pooling layer

The pooling layer was proposed by Boureau et al. (2010) to create the feature maps in which the size of the feature maps was reduced after applying the pooling operation. In this study, I applied the average pooling layer so that a small translation of the input image does not affect the output values. The pooling operation is regularly applied after a convolutional layer. In order to create feature maps, it calculates the average value of pixels in each area of a feature map. Further, I aim to decrease both the number of CNN parameters and the computational time. The average pooling layer is calculated as follows.

$$f_{ave}(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (6)$$

where x is the vector containing the pixel values from the local pooling region, N is the number of pixels. Typically, the size of the pooling operation is 2x2 or 3x3 blocks. In our network, an average pooling layer of size 3x3 was applied.

2. Global average pooling layer

Global average pooling (GAP) (Lin et al., 2014) was introduced to replace the traditional fully connected layers in the CNN architecture. Hence, the output of the GAP layer is given directly to the softmax layer. The purpose of the GAP layer is to calculate each corresponding feature map by averaging the values of the corresponding feature map and transforming it into only one feature. For example, the feature map size of 3x3x2048 would be output as 1x1x2048. In the GAP layer, it does not have a parameter to optimize. The spatial information of feature maps is averaged,

more robust to spatial translations from the input feature maps. Consequently, the overfitting problem is avoided at this layer.

4.3.2 Spatial Feature Extraction Network

In this section, I propose a spatial feature extraction network, as shown in Figure 28, to extract the spatial features from the food images. According to experimental results given in Phiphiphatphaisit and Surinta (2020), I chose the ResNet50 architecture that reached the best performance on the benchmark ETH Food-101 dataset. First, the input images were resized to the fixed size of 224x224 pixels with three channels that fit the input layer of the ResNet50. Second, the last pooling layer of ResNet50 was decreased by applying the reduction operation. Finally, the convolutional 1D (Conv1D) block was attached to the reduction operation. The output of the Conv1D block was the robust spatial features.

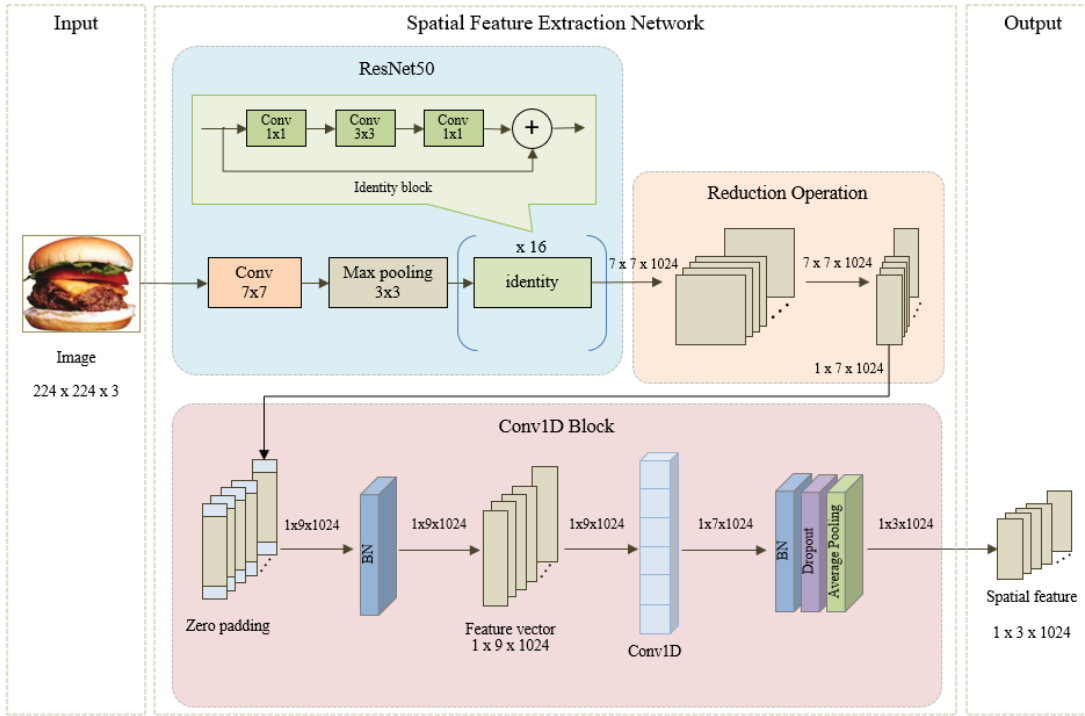
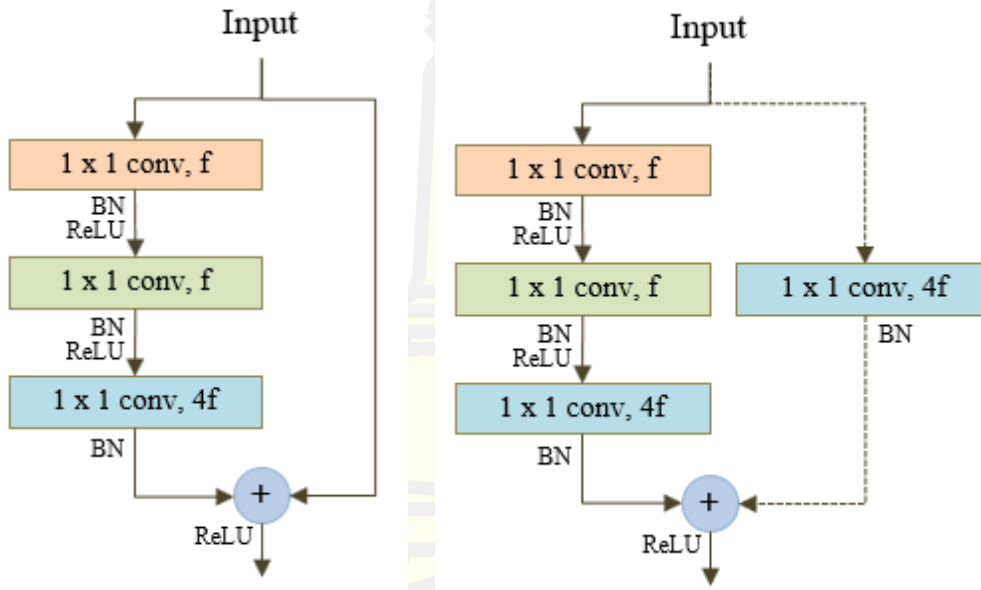


Figure 28 *Illustrated Spatial Feature Extraction Network*

4.3.2.1 ResNet

Residual Network (ResNet) (He et al., 2016) is the deep convolutional network using shortcut connection, namely residual block, that allows each layer to skip over one or more layers. Residual block typically contains a batch

normalization layer (BN) and ReLU activation function. Further, BN is attached after each convolutional layer and followed by the ReLU function. The residual block follows two simple rules: 1) when the input from the previous residual block and output of the current residual block is presented as the same dimension, called identity mapping, it takes outputs from the previous block and adds with the output from the skipped layers, as shown in Figure 29a. 2) When the input of the previous residual block and output of the current residual block are not the same size, the projection shortcut is implemented to ensure that the output of the residual block is the same size after applying the addition operation, as shown in Figure 29b.



(a) (b)

Figure 29 *Bottleneck block for ResNet50: (a) identity shortcut, (b) projection shortcut. (f denotes the number of filters)*

In this experiment, I trained the model using a pre-trained model of ResNet50 to speed up the training process. However, I removed the fully connected and extracted the spatial from the last layers of the ResNet50.

4.3.2.2 Reduction Operation

I implemented the reduction operation that aimed to adjust the size of the feature maps. The size of the feature maps that were extracted using the ResNet50 was defined by the three dimensions (width x height x number of feature

maps). Hence, the input layer of the convolutional 1D block should be in the form of two dimensions. In our study, the reduction operation was installed between the ResNet50 and Conv1D block. The reduction operation is calculated as follows Equation.

$$Fv_i = cat\left(max\left(x_j\right)\right), j\epsilon\{1,2,\dots,k\} \qquad (4)$$

where Fv is the feature vector when $Fv_i\epsilon\{Fv_1, Fv_2, \dots, Fv_i\}$, i is the number of feature maps, x_j is the vector of a region with the size of 1 x H when H denotes the height of the feature vector in each feature map, cat is concatenate the maximum value of x_j, when $j \in \{1,2,\dots,k\}$ and k denotes the width of the feature vector in each feature map.

4.3.2.3 Convolutional 1D Block

In our proposed Conv1D, the spatial features extracted using the ResNet50 architecture were first given to the reduction operation to transform the feature maps into one dimension. Second, I computed the zero-padding operation to the spatial features, followed by the BN operation. Then, the 1D convolution operation with a filter size of 1x3 and a stride of 1 was calculated through the spatial features after applying zero padding. Third, three operations; BN, dropout, and average pooling, were attached to the network. Finally, the robust spatial features were obtained from the spatial feature extraction network, as shown in Figure 28.

4.3.3 Temporal Feature Extraction Network

This section investigated the long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) to extract the robust temporal features. The LSTM network was proposed to learn patterns in long sequence data by combining cell state and three gates; input, output, and forget. In the LSTM network, the cell state function is to provide relevant sequence information into gates. The gates in the LSTM network are chosen which information is allowed and which information is related to keep or forget while training.

This study applied the LSTM network to learn the sequence data extracted using the spatial feature extraction network described in Section 4.3.2. The temporal feature is the output of the LSTM network, as shown in Figure 30.

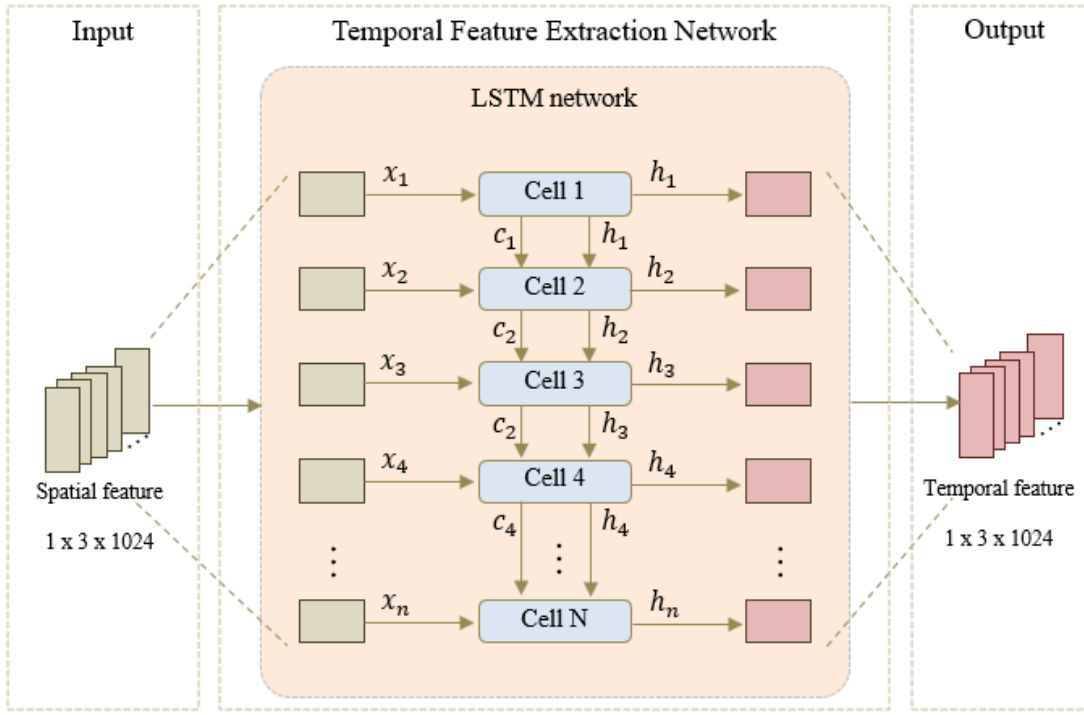


Figure 30 *Illustration of the LSTM network proposed to extract the temporal features.*

4.3.3 Adaptive Feature Fusion Network

I proposed an adaptive feature fusion network that combines robust spatial-temporal feature networks extracted from the spatial feature extraction network (see Section 4.3.2) and the temporal feature extraction network (see Section 4.3.3), as shown in Figure 31. Furthermore, after concatenating two robust features, the robust features were given to the GAP layer, followed by the BN layer. The ReLU activation function was calculated while training. Finally, the robust feature vector was classified using the softmax function. The details of the adaptive feature fusion network are shown as follows.

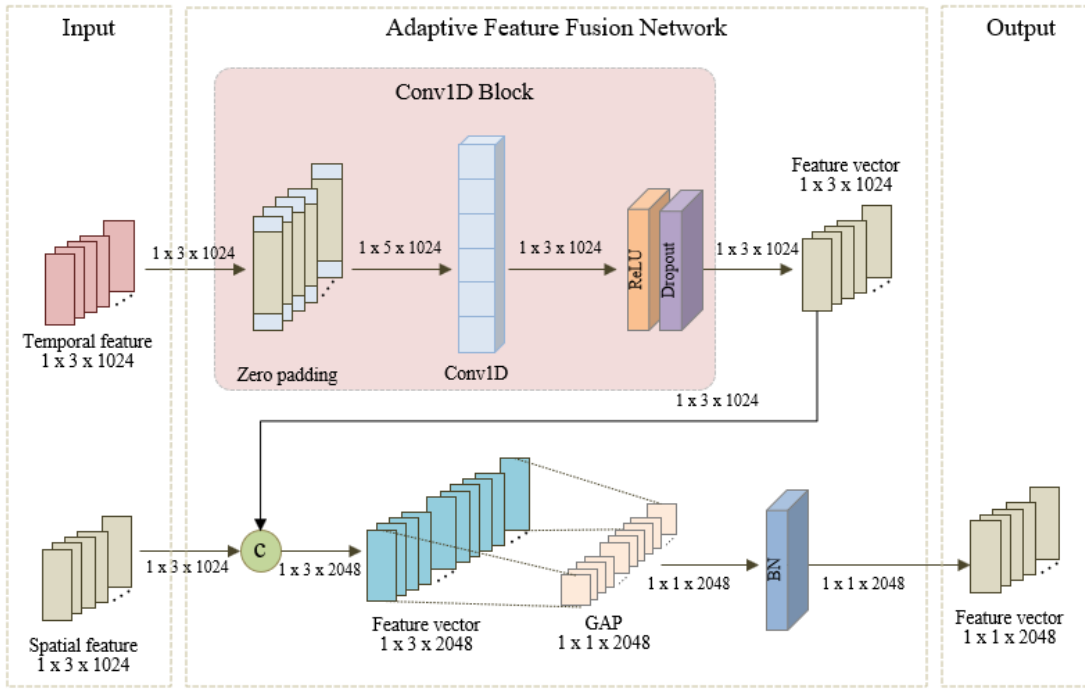


Figure 31 *Illustrated Adaptive Feature Fusion Network*

4.3.3.1 Convolutional 1D Block

In this section, the Conv1D block was different from the Conv1D in Section 4.3.2.3. First, the input of the Conv1D block was the temporal feature extracted using the LSTM network. Second, the 1D convolution operation with a filter size of 1x3 and a stride of 1 was computed. Finally, the ReLU activation function and Dropout layer were combined to the last layer of the Conv1D block.

4.3.3.2 Concatenate between Conv1D Block and Spatial Features

The last step of the adaptive feature fusion network was that the output of the Conv1D block and spatial features from the spatial feature extraction network (Section 4.3.2) were concatenated. In addition, the GAP and the BN layers were invented to decrease the network parameters and standardize the feature vector before assigning the features to classify with the softmax function.

4.4 Real-World Food Image Datasets

I evaluated our proposed adaptive feature fusion network (ASTFF-Net) on four benchmark food image datasets, including Food11, UEC Food-100, UEC Food-256, and ETH Food-100. The details of each food image dataset were as follows:

4.4.1 Food11 Dataset

Singla et al. (2016) proposed the Food11 dataset that consisted of 16,643 food images of 11 categories that were bread, dairy products, egg, dessert, meat, fried food, pasta, seafood, rice, vegetables/fruit, and soup, as shown in Figure 32.

Figure 32 *Sample images of the Food11 datasets*

4.4.2 UEC Food-100 Dataset

Matsuda et al. (2012) collected the UEC Food-100 dataset. It contains 14,361 images from 100 categories of famous Japanese foods, such as sushi, eels on rice, pilaf, beef curry, fried noodle, and tempura. The UEC Food-100 dataset consists of multiple food items in one image (see Figure 33a) and a single food item in one image (see Figure 33b).

(a) (b)

Figure 33 *Examples of the UEC Food-100 dataset, (a) Multiple food items and (b)single food items.*

4.4.3 UEC Food-256 Dataset

Kawano & Yanai (2014) proposed the UEC Food-256 image dataset, which is the extended version of the UEC Food-101 dataset. First, all the images were collected from Flickr, Bing, and Twitter, using a specific query. Second, the downloaded images were classified using the Foodness method and categorized as food or non-food images. Finally, the UEC Food-256 dataset contained approximately 32,000 food images and comprised 256 categories with more than 600 food images in each category after removing noise images. Examples of the UEC Food-256 dataset are shown in Figure 34b.

(a) (b)

Figure 34 *Illustration of (a) the ETH Food-101 dataset (b) the UED-Food256 dataset.*

4.4.4 ETH Food-101 Dataset

The ETH Food-101 dataset was proposed by Bossard & Gool (2014), which is the real-world food images downloaded from the website foodspotting.com. It contains 101,000 food images and has 101 food image categories. The examples of the ETH Food-101 dataset are shown in Figure 34a.

The summary details of four benchmark food image datasets are shown in Table 12.

Table 12 *Illustrated the details of the benchmark food image datasets.*

Datasets	Categories	No. of Images	No. of Training	No. of Testing	Image per Category
Food11	11	16,643	12,483	4,160	Unbalanced
UEC Food-100	100	14,361	10,771	3,590	Unbalanced
UEC Food-256	256	31,395	23,547	7,848	Unbalanced
ETH Food-101	101	101,000	75,750	25,250	1,000

4.5 Experimental Results and Discussion

In this section, I implemented the adaptive feature fusion network with the TensorFlow platform running on Google Colab with GPU support for all the experiments. The proposed adaptive spatial-temporal feature fusion network (ASTFF-Net) was evaluated on various benchmark food image datasets, including Food11, UEC Food-100, UEC Food-256, and ETH Food-101. I divided the food image datasets into training and test sets. The accuracy of the ASTFF-Net was evaluated on the test set. Moreover, I employed 5-fold cross-validation (cv) over the training set to find the significance of the proposed network and prevent overfitting problems. The average accuracy, standard deviation, recall, and F1-score were reported.

In the ASTFF-Net, I used only the pre-trained model of the ResNet50 architecture with pre-trained weights from the ImageNet dataset. However, other parts of the framework do not transfer from the pre-trained model. I trained the ASTFF-Net with the SGD optimizer to optimize the loss function. The adaptive learning rate was proposed with the initial value of 0.01 and then reduced to 0.0001 when the loss value did not decrease after five epochs. The momentum value was set to 0.9 and the weight decay was updated based on the learning rate value and the number of epochs. The ASTFF-Net was trained for only 50 epochs.

To study the efficiency of the ASTFF-Net, I invented four different experiments. First, I combined spatial and temporal features, called the ASTFF-NetB1 model, as shown in Figure 35a. Second, the spatial features were sent to the Conv1d block before combining with the temporal features, called the ASTFF-NetB2, as shown in Figure 35b. Third, the temporal features were sent to the Conv1D block before combining with the spatial features, called the ASTFF-NetB3, as shown in Figure 35b. Finally, both spatial and temporal features were given to the Conv1D block before combining, called the ASTFF-NetB4, as shown in Figure 35d.

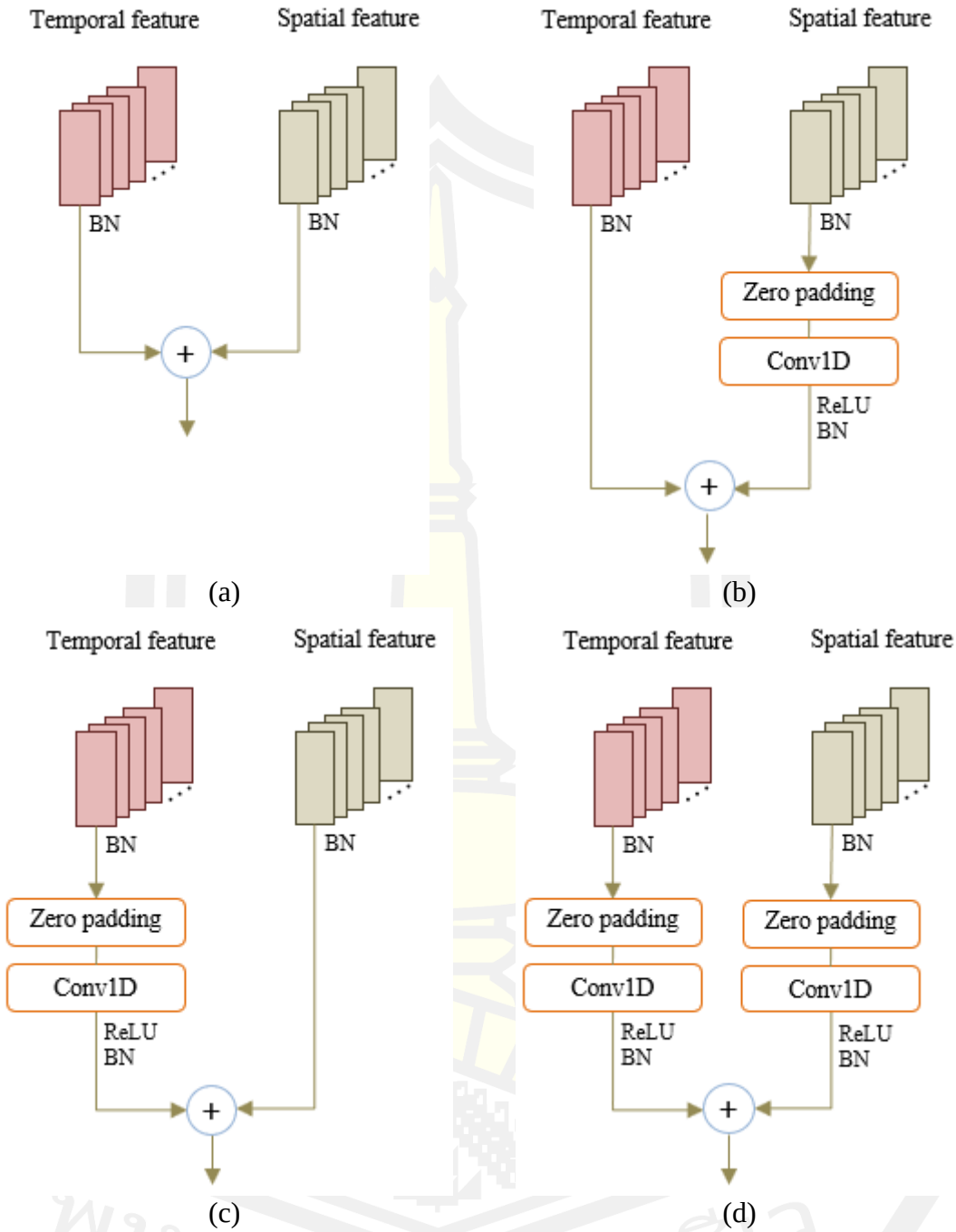


Figure 35 *Illustration of ASTFF-Nets used in the experiments. (a) the ASTFF-Net baseline network, called ASTFF-NetB1, (b) ASTFF-NetB2, (c) ASTFF-NetB3, and (d) ASTFF-NetB4.*

4.5.2.1 Experiments on the Food11 Dataset

I trained four ASTFF-Nets on the Food11 dataset on the training data based on five-fold cross-validation (5-cv) and evaluated ASTFF-Net models on a separate test set. The results obtained are presented in Table 13.

Table 13 *Evaluation performances (average accuracy, ± standard deviation, test accuracy, recall, and F1-score) of the ASTFF-Nets on the Food11 dataset. The bold numbers represent the best ASTFF-Net model.*

Model	5-CV	Accuracy (%)	Recall	F1-Score	Testing Time
ASTFF-NetB1	94.26 ± 0.177	93.47	0.935	0.935	5m:22s
ASTFF-NetB2	94.17 ± 0.291	93.16	0.932	0.932	5m:24s
ASTFF-NetB3	**96.08 ± 0.330**	**95.04**	**0.950**	**0.950**	5m:24s
ASTFF-NetB4	95.54 ± 0.369	94.63	0.946	0.946	5m:25s

From Table 13, I observed that ASTFF-NetB3, in which the temporal features were sent to the Conv1D block before combining with the spatial features, outperformed other ASTFF-Nets on the Food-11 image dataset. The ASTFF-NetB3 achieved 96.08% accuracy on the training set using 5-cv and 95.04% accuracy on the test set, which was the best network. On the other hand, ASTFF-NetB2 had the worst performance on both training and test sets. However, it was only approximately 1.8% below that of ASTFF-NetB3. Further, as for the testing time, all ASTFF-Nets performed almost a similar computation. It spent approximately 5 minutes on the whole test set (approximately 75.28 milliseconds per image).

Figure 36 illustrates the confusion matrix of four ASTFF-Nets. It was found that the ASTFF-NetB3 (see Figure 36c) reduced the misclassified number of images from category egg to bread. It reduced the misclassified images from 17 images to only two images. Also, the rice category that was misclassified to the fruit/veg category was reduced from 4 images to zero.

Figure 37 shows the probability of the egg (see Figure 37a) and rice (see Figure 37b) categories that were classified using ASTFF-NetB3, but other ASTFF-Nets misclassified it.

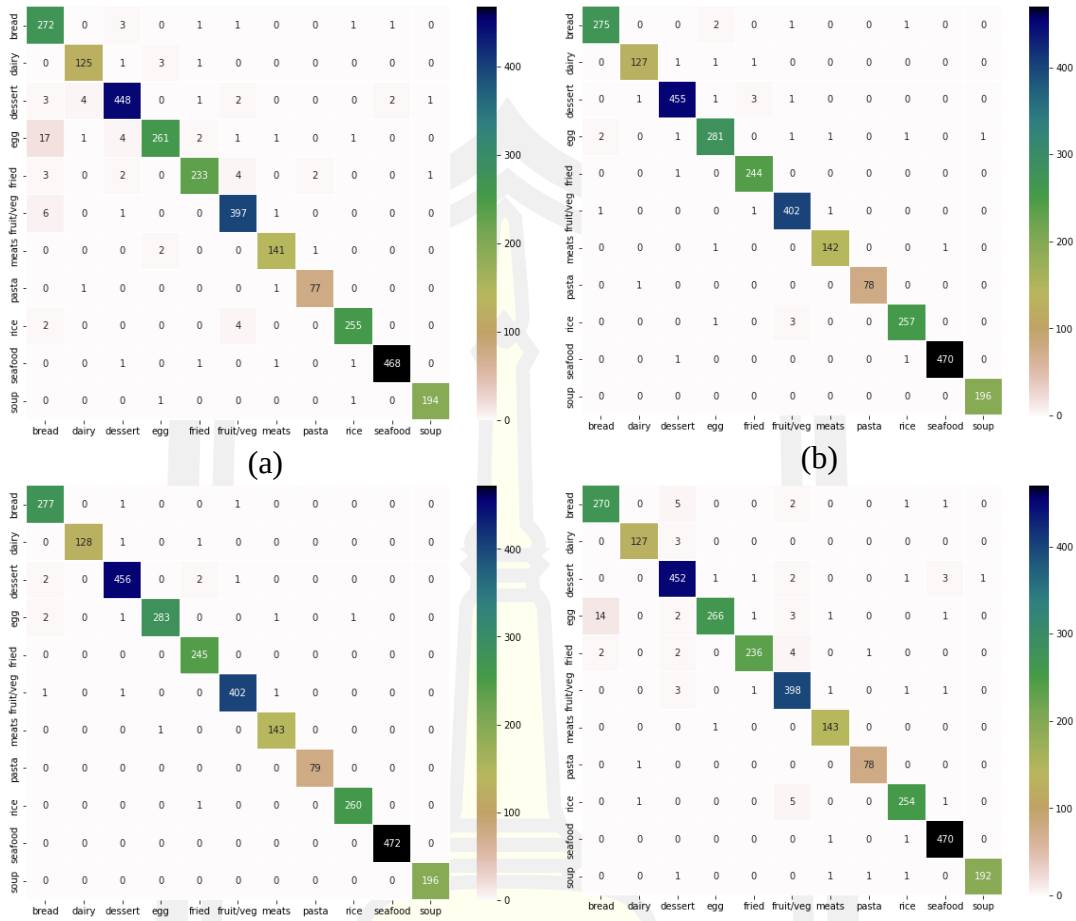


Figure 36 *Illustration of confusion matrix of ASTFF-Net on Food11 datasets, (a) ASTFF-NetB1, (b) ASTFF-NetB2, (c) ASTFF-NetB3, (d) ASTFF-NetB4.*

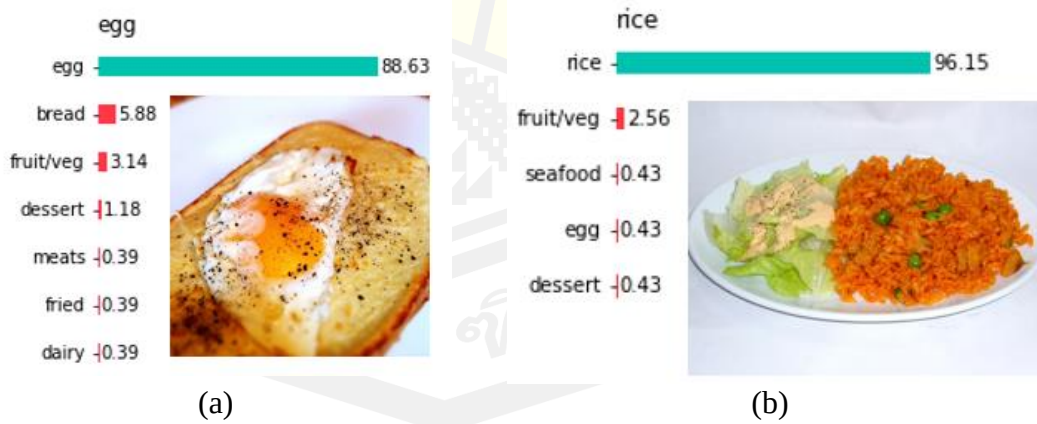


Figure 37 *Example of Food11 classes which are misclassified based on confusion matrix generated from ASTFF-NetB3.*

Table 14 *Recognition performance of the Food11 dataset when compared with different deep learning techniques.*

References	**Methods**	**Test Accuracy (%)**
McAllister et al. (2018)	ResNet152+ANN	91.34
Akbulut, & Budak (2019)	AlexNet+VGGl6+SVM	88.08
Our Proposed	ASTFF-NetB1	93.47
	ASTFF-NetB2	93.16
	ASTFF-NetB3	**95.04**
	ASTFF-NetB4	94.63

I present extensive comparisons of our ASTFF-Nets on the Food11 dataset with existing state-of-the-art methods, as shown in Table 10. The experimental results confirm that our ASTFF-Nets increase the accuracy performance. Additionally, our ASTFF-Nets show much better results than extracting the deep features using CNN architectures and combining them with machine learning techniques, such as artificial neural networks and support vector machines (McAllister et al., 2018 Akbulut & Budak, 2019). In conclusion, the ASTFF-NetB3 results in the highest accuracy performance of 95.04%.

4.5.2.2 Experiments on the UEC Food-100 Dataset

This section showed that our ASTFF-Nets also present the best accuracy performance on the UEC Food-100 dataset, which has 100 food categories. The results achieved throughout the testing process are shown in Table 15.

Table 15 *Evaluation of the classification results for the UEC Food-100 dataset using different ASTFF-Net method.*

Model	**5-CV**	**Accuracy (%)**	**Recall**	**F1-Score**	**Testing Time**
ASTFF-NetB1	86.77 ± 0.231	85.70	0.857	0.857	4m:38s
ASTFF-NetB2	86.99 ± 0.267	86.05	0.861	0.861	4m:39s
ASTFF-NetB3	$\mathbf{92.55 \pm 0.168}$	**91.35**	**0.914**	**0.914**	4m:39s
ASTFF-NetB4	89.85 ± 0.344	88.85	0.889	0.889	4m:41s

From Table 15, the results showed that ASTFF-NetB3 significantly outperforms other ASTFF-Nets on the UEC Food-100 dataset (t-test, $p < 0.05$). I observed that the ASTFF-NetB3 performed with higher than 4% accuracy on

the 5-cv and higher than 5% accuracy on the test set when compared with other ASTFF-Nets. Another observation is that the ASTFF-NetB3 achieved an F1-score of more than 0.90, which means that the ASTFF-NetB3 successfully classified food images over a specific strength with a low false-positive rate. Moreover, all the ASTFF-Net architectures still spent fast on the test set with approximately 73.20 milliseconds per food image.

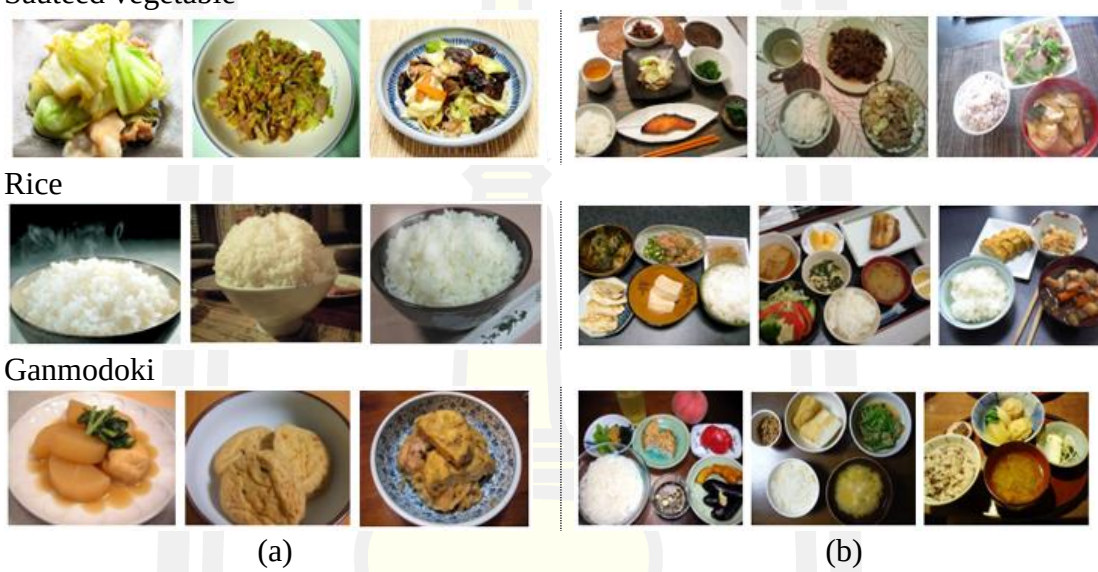


Figure 38 *Some examples of sauteed vegetables, rice, and ganmodoki images of the UEC Food-100 dataset were classified using the ASTFF-NetB3 model. The food images were (a) correctly classified and (b) misclassified.*

I illustrated the food images that were correctly classified when using the ASTFF-NetB3 model, as shown in Figure 38a. All the food images contained only one dish, which means only one food category appeared in the image. On the other hand, the mostly misclassified food images, as shown in Figure 38b, always included many objects in one image. For example, the rice dish appears in sauteed vegetables and ganmodoki categories.

Table 16 *Recognition performance of the UEC Food-100 dataset when compared with different deep learning techniques.*

References	**Methods**	**Test Accuracy (%)**
Liu et al. (2016)	DeepFood	76.30
Hassannejad et al. (2016)	InceptionV3	81.45
Martinel et al. (2018)	WISeR	89.58
Tasci (2020)	Ensemble CNNs	84.52
Our Proposed	ASTFF-NetB1	85.70
	ASTFF-NetB2	86.05
	ASTFF-NetB3	**91.35**
	ASTFF-NetB4	88.85

Table 16 compares the performance of our approach architectures on the UEC Food-100 dataset with existing deep learning techniques. The accuracy performance of the previous deep learning techniques did not achieve very high scores, even using the ensemble CNNs method (Tasci, 2020). The highest accuracy was not above 90% with the WISeR method (Martinel et al., 2018). However, the ASTFF-NetB1, B2, and B4 did not achieve higher performance than the WISeR method. Consequently, the proposed ASTFF-NetB3 network, that directly gives the temporal feature to the Conv1D block and then combines it with the spatial features, demonstrated the highest performance with 91.35% accuracy.

4.5.2.3 Experiments on the UEC Food-256 Dataset

In this section, I evaluated the proposed adaptive network on the UEC Food-256 dataset in terms of 5-cv, test accuracy, recall, and F1-score. It has a huge category with 256 menus from Japan and other countries. The proposed ASTFF-Nets were evaluated on 23,547 training images and 7,848 test images.

Table 17 *Evaluation of the classification results for the UEC Food-256 dataset using different ASTFF-Net method.*

Model	**5-CV**	**Accuracy (%)**	**Recall**	**F1-Score**	**Testing Time**
ASTFF-NetB1	92.16 ± 0.192	91.07	0.911	0.911	10m:08s
ASTFF-NetB2	92.05 ± 0.155	90.90	0.909	0.909	10m:11s
ASTFF-NetB3	$\mathbf{93.21 \pm 0.324}$	**92.15**	**0.921**	**0.921**	10m:11s
ASTFF-NetB4	92.40 ± 0.301	91.37	0.914	0.914	10m:14s

Table 17 shows the evaluation performance of the ASTFF-Nets. I observed that the ASTFF-NetB3 consistently achieved the highest accuracy and significantly outperformed other ASTFF-Nets (t-test, $p < 0.05$) on both 5-cv and test sets. The ASTFF-NetB2 slightly decreased the performance on the UEC Food-256 dataset. Consequently, our proposed ASTFF-Nets achieved above 90% accuracy. It spent approximately 10 minutes on the whole test set (approximately 77.29 milliseconds per image).

As illustrated in Figure 39, I discovered that some food images have similar texture, color, and pattern characteristics that could harm the proposed ASTFF-Nets to misclassification.

Figure 39 *Illustration of the similar food images between (a) ramen noodle and tensin noodle, (b) raisin bread and cream puff, and (c) egg sunny side up and green curry.*

Table 18 *Recognition performance of the UEC Food-256 dataset when compared with different deep learning techniques.*

References	**Methods**	**Test Accuracy (%)**
Liu et al. (2016)	DeepFood	54.70
Hassannejad et al. (2016)	InceptionV3	76.17
Martinel et al. (2018)	WISeR	83.15
Tasci (2020)	Ensemble CNNs	77.20
Our proposed	ASTFF-NetB1	91.07
	ASTFF-NetB2	90.90
	ASTFF-NetB3	**92.15**
	ASTFF-NetB4	91.37

Table 18, I observed that the existing deep learning methods did not show high accuracy. The WISeR method (Martinel et al., 2018) achieved the

best performance with an accuracy of only 83.15%. The proposed ASTFF-Nets performed much better than the previous methods and achieved above 90% accuracy. Consequently, the ASTFF-NetB3 always achieved the best performance with an accuracy of 92.15%, which is approximately 9% over the WISeR method.

4.5.2.4 Experiments on the ETH Food-101 Dataset

In this experiment, I tested the proposed adaptive network on the ETH-Food101 dataset, which has 75,750 training images and 25,250 test images. It is the largest food image dataset that I evaluated in our experiments. The results of the proposed ASTFF-Nets are shown in Table 19.

Table 19 *Evaluation of the classification results for the ETH-Food101 dataset using different AFF-Net method.*

Model	5-CV	Accuracy (%)	Recall	F1-Score	Testing Time
ASTFF-NetB1	91.88 ± 0.229	91.13	0.911	0.911	32m:35s
ASTFF-NetB2	90.16 ± 0.276	89.05	0.890	0.890	32m:45s
ASTFF-NetB3	**93.98** ± 0.247	**93.06**	**0.931**	**0.931**	**32m:45s**
ASTFF-NetB4	93.56 ± 0.224	92.81	0.928	0.928	32m:55s

Table 19 reports that the ASTFF-NetB3 still achieved the best performance when compared with other ASTFF-Nets (t-test, $p < 0.05$, significant). It achieved a performance of 93.98% accuracy on 5-cv and 93.06% accuracy on the test set. Furthermore, I found that the ASTFF-NetB3 achieved the highest accuracy on four food image datasets; ETH Food-101, Food11, UEC Food-100, and UEC Food-256. Considering the computational time, all the ASTFF-Net architectures spent approximately 77.11 milliseconds per food image on the test set.

I also observed that ASTFF-NetB3 achieved an F1-score of 0.931 with a high true-positive rate. The illustration of the F1-Score, when classified using the ASTFF-Nets, is shown in Figure 40. Moreover, for further investigation, I found noise and non-food objects in some food categories, such as apple pie and Peking duck. The example of the noise and non-food objects is shown in Figure 41.

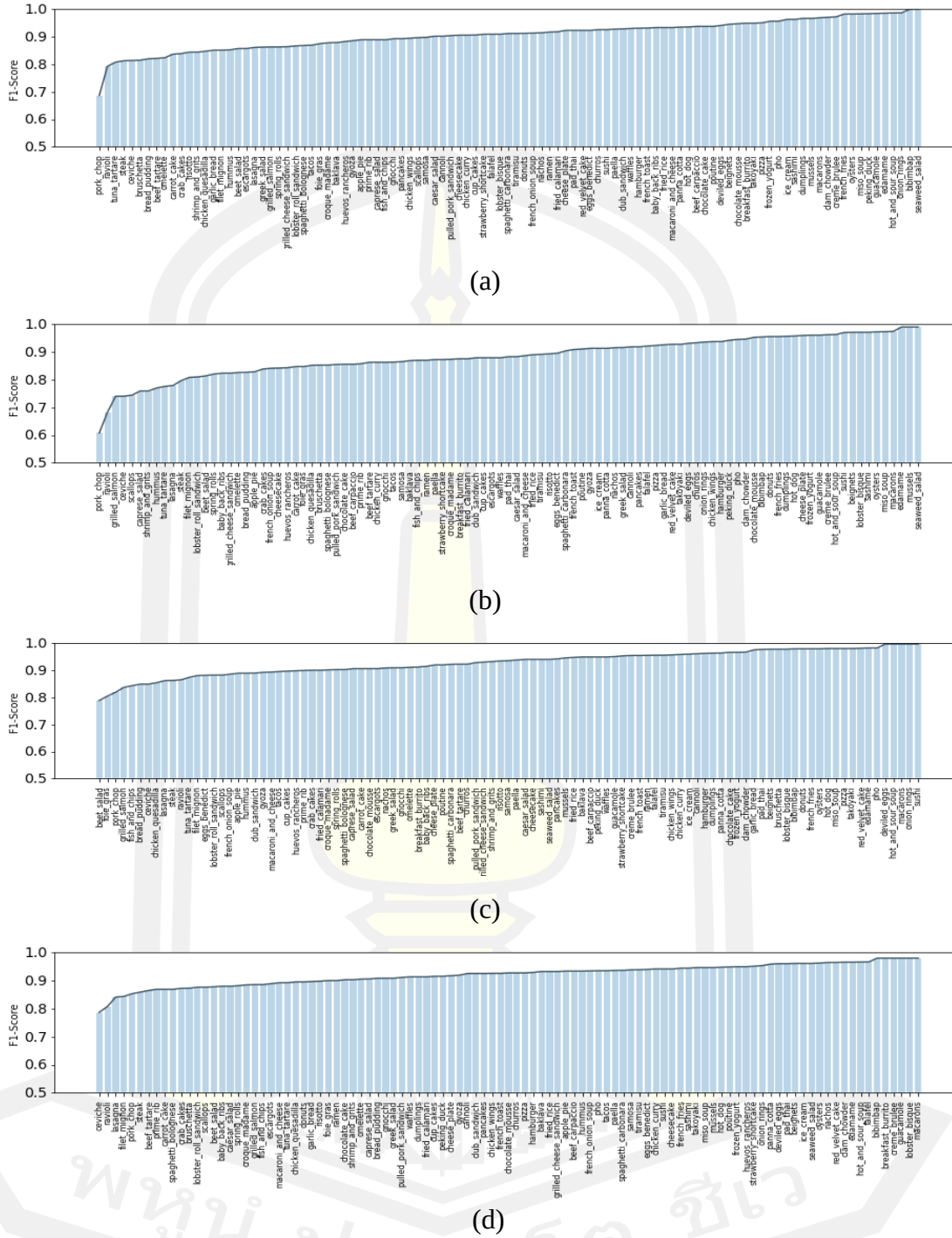
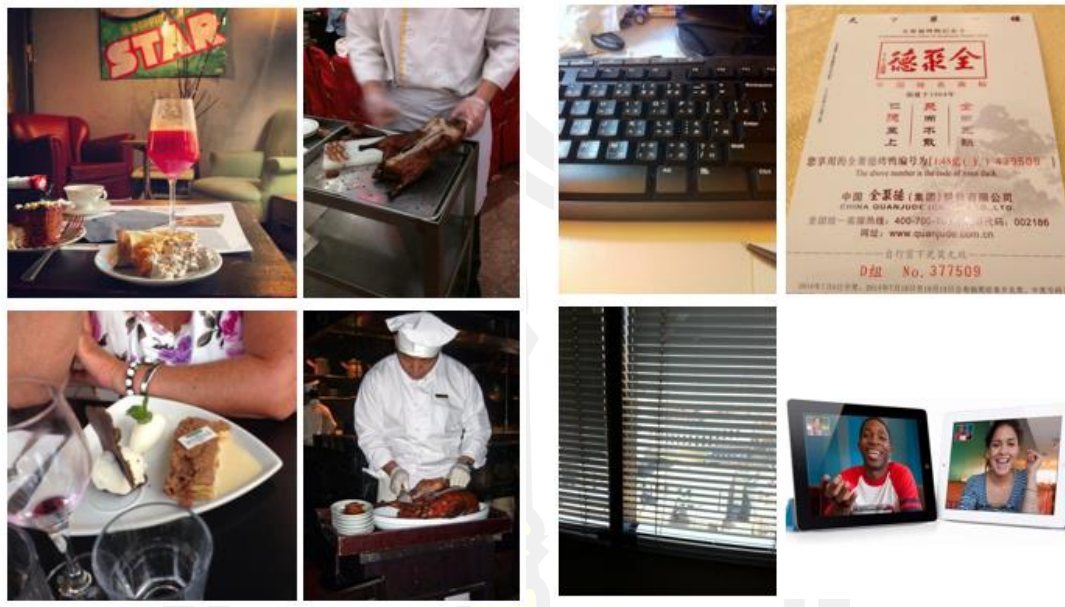


Figure 40 *Illustration of F1-Score using the ASTFF-Net models to classify ETH Food-256 dataset. (a) ASTFF-NetB1, (b) ASTFF-NetB2, (c) ASTFF-NetB3, (d) ASTFF-NetB4.*



(a) (b)

Figure 41 *example of the noise and non-food objects. (a) noise in food image, (b) non-food objects.*

Table 20 *Recognition performance of the ETH-Food101 dataset when compared with different deep learning techniques.*

References	**Methods**	**Test Accuracy (%)**
Liu et al. (2016)	DeepFood	77.40
Hassannejad et al. (2016)	InceptionV3	88.25
Bolanos and Radev (2016)	GoogLeNet	79.20
Pandey et al. (2017)	EnsembleNet	72.12
Aguilar et al. (2017)	CNNs Fusion	86.71
Martinel et al. (2018)	WISeR	90.27
McAllister et al. (2018)	ResNet152	64.98
Akbulut, & Budak (2019)	AlexNet+VGGl6+SVM	79.86
Tasci (2020)	Ensemble CNNs	84.28
Phiphiphatphaisit & Surinta (2020)	Modified MobileNetV1	72.59
Phiphiphatphaisit & Surinta (2021)	ResNet50+Conv1D-LSTM	90.87
Our proposed	ASTFF-NetB1	91.13
	ASTFF-NetB2	89.05
	ASTFF-NetB3	**93.06**
	ASTFF-NetB4	92.81

In table 20, I compared the proposed ASTFF-Nets with other methods. I observed that extraction the deep features using CNN, Conv1D, and LSTM (Phiphiphatphaisit & Surinta, 2021) performed better than training with only CNN architectures and even better than extracting the deep features and combined with machine learning techniques. The results in Table 20 show that our ASTFF-NetB1, B3, and B4 were given an accuracy above 90%. These networks also outperformed various existing methods. Consequently, the ASTFF-NetB3 achieved an accuracy of 93.06%, which is the highest performance on the ETH Food-101 dataset.

4.5.3 Discussion

In this research, I discussed four important issues that affect the performance of the CNN models.

Overfitting with Robust Network: Naturally, overfitting problems occur when very deep CNN layers are proposed to create the robust CNN model and also trained with too many example images. With very deep CNN architectures, the CNN model actually needs to optimize many hyperparameters. To face this problem, I proposed the adaptive spatial-temporal feature fusion network, called ASTFF-Net, which was invented to combine both spatial and temporal feature extraction networks. The adaptive architectures were designed to extract information on the spatial domain and ignore some insignificant information using the temporal network. I evaluated the proposed method using a five-fold cross-validation method (5-cv), as shown in Table 15, and I found that the ASTFF-Nets could learn well with many training examples and generalize well with the test set. The 5-cv and the test set results were not given an enormous difference.

Similarity patterns between two categories: The real-world food images from the benchmark datasets were downloaded from the internet Some of the images contain many noise objects (see Figure 41a), some images have similar patterns (see Figure 39b) and some images contain similar food objects (see Figure 39c) that appear in many food categories. For example, the category of the bread dish was classified as the egg category because the bread is actually served with egg. I then presented the F1-Score to measure the precision of the ASTFF-Net architecture.

Furthermore, the confusion matrix, as shown in Figure 36c, confirmed that the ASTFF-NetB3 can address the similarity pattern between two classes; egg and bread.

Multi-object Problem: The UEC Food-100 dataset usually contains the multi-object appearing in one image, as shown in Figure 33a. It is not easy to recognize as the correct category because many dishes are included in the image. As a result, it is misclassified. With the multi-object problem, I carefully checked the recognition results of the proposed ASTFF-Nets and found that the proposed network recognized one correct dish from many dishes that appear in one image. For example, the image contains fish, soup, rice, and sauteed vegetables in the sauteed vegetable category. So, the ASTFF-NetB3 was classified as rice, which was one category from many categories from the image. To address the multi-object problem, thus, I recommend applying object detection and classifying each object.

Computational cost and Model size: I designed the ASTFF-Nets according to the advantage of extracting the spatial and temporal deep features. Further, three networks were included in the ASTFF-Nets; spatial feature extraction, temporal feature extraction, and adaptive feature fusion. Indeed, the ASTFF-Nets had a larger model size than the CNN and CNN-LSTM networks, as shown in Table 21. However, when I evaluated the proposed ASTFF-Nets on the test set, the computation cost of the ASTFF-Nets did not significantly increase. It increased only around four milliseconds and only 0.6 milliseconds compared with the ResNet50 and CNN-LSTM respectively. The comparison of the model size and testing time is shown in Table 21.

Table 21 *The comparison of the computational cost and model size between the proposed ASTFF-Nets and other architectures.*

Methods	**Testing Time (~ms/im.)**	**Size (M)**
ResNet50	73.2	24.6
ResNet50+Conv1D-LSTM (Phiphiphatphaisit & Surinta, 2021)	77.2	38.3
ASTFF-NetB1	77.4	38.4
ASTFF-NetB2	77.8	41.5
ASTFF-NetB3	77.8	41.5
ASTFF-NetB4	78.2	44.7

4.6 Conclusions

In this research, the adaptive spatial-temporal feature fusion network, namely ASTFF-Net, was invented to improve the food image recognition performance. In other food recognition systems, a convolutional neural network (CNN) is usually proposed to extract the spatial features from the food images. However, real-world food images sometimes contain many noise and non-food objects, resulting in the CNN extracting deep features containing information of the object mentioned. Consequently, I proposed to use ResNet50 to extract the spatial features and directly send them to the convolutional 1D (Conv1D) block, followed by a long short-term memory (LSTM) network. The LSTM network has gate operations designed to learn sequence patterns from spatial information and allow which information to keep or forget during the training scheme. The ASTFF-Net architecture is divided into three parts as follows. First, the spatial feature extraction network, I proposed to use the state-of-the-art CNN model, namely ResNet50, to extract temporal features. Then, the reduction operation was attached to the ResNet50 to minimize the size of the feature maps before sending them to the Conv1D block. Second, the temporal feature extraction network, the sequence output of the Conv1D block was assigned to the LSTM network to create temporal features. Third, the spatial and temporal features from the first and second parts were combined using concatenation operation, then assigned to the Conv1D, called adaptive feature fusion network. As with the ASTFF-Net, the softmax function was connected to the ASTFF-Net as the recognition layer proposed to recognize real-world food images. The ASTFF-Net architecture was proposed to address the overfitting problems because I combined the global average pooling (GAP) and dropout layers to the architecture. The most benefit of the GAP layer is that the parameter of the ASTFF-Net was reduced. Additionally, the unnecessary connections between layers were dropped using the dropout layer.

In the experiments, I evaluated four ASTFF-Nets on four different real-word food image datasets: Food11, UEC Food-100, UEC Food-256, and ETH Food-101. The results show that the ASTFF-Nets achieved the highest accuracy on 5-cv and the test set. Furthermore, I found that the proposed ASTFF-NetB3 outperformed the existing methods on four food image datasets.

In future research, I will apply the ASTFF-Nets to address the challenge of the unbalanced datasets (Aggarwal, Popescu, & Hudelot, 2020). Another direction will be applying the instance selection methods (Branikas, Papastergiou, Zacharaki, & Megalooikonomou, 2019) to reduce the training set. It may reduce the training set by more than 50%. Then, the computational time will decrease while training the ASTFF-Nets. Finally, I will consider the segmentation techniques (Hafiz & Bhat, 2020) that can select the most relevant food region from the real-world food images.

Chapter 5
Discussion

The objective of this thesis is to propose novel deep learning approaches to address the problems of food image recognition. Firstly, I proposed a new convolutional neural network (CNN) based on MobileNet architecture that decreased the parameters of the CNN model. I also concentrated on reducing the training data size and proposed using data augmentation techniques to increase the variance of training data and prevent overfitting on the test set. Secondly, the robust deep feature extraction method based on convolutional 1D (Conv1D) and long short-term memory (LSTM) was evaluated on a food image dataset with 101 food categories. Thirdly, to overcome the advantage of the Conv1D-LSTM network, an adaptive feature fusion network, called ASTFF-Net, was proposed. This network was designed to extract the robust deep features that were extracted using Conv1D and LSTM networks. Consequently, I have performed the proposed ASTFF-Net on four real-word food image datasets; Food-11, UEC Food-100, UEC Food-256, and ETH Food-101.

I will now briefly describe and discuss the challenges of the food image recognition systems using a deep learning approach.

Chapter 2 showed that, due to the difficulties of real-world food images, food images can be taken from different perspectives and many objects can also appear in the food image. To solve this challenge, I proposed a new CNN model that was modified from the state-of-the-art MobileNet architecture. Our modified MobileNet network decreased the parameters of the CNN model, but still achieved high accuracy. In the modified MobileNet, I ignored the average pooling layer and the fully connected layer (FC), and replaced them with the global average pooling layer (GAP) followed by the batch normalization layer (BN) and rectified linear unit (ReLU) activation function. I also considered avoiding overfitting by combining the dropout layer after the ReLU function. I also performed the data augmentation techniques to avoid overfitting, including rescaling, rotation, width shift, height shift, horizontal flip, shear, zoom, and random cropping. I performed experiments on a

publicly available dataset, called the ETH food-101 dataset. The experimental results showed that the modified MobileNet architecture improved accuracy by approximately 5% when training with the data augmentation.

In Chapter 3, I mainly concentrated on extracting the robust feature using the deep feature extraction technique. Firstly, I proposed to use CNN architectures to extract the deep features from the food images, called the spatial features. Secondly, I then transferred the spatial features into the LSTM network to extract the temporal feature. Thirdly, the deep features were extracted using a 1D convolutional and LSTM network called Conv1D-LSTM. Finally, the deep features were classified using the softmax function. To extract the robust spatial features, I proposed six state-of-the-art CNN architectures, consisting of VGG16, VGG19, ResNet50, DenseNet201, MobileNetV1, and MobileNetV2.The transfer learning method was proposed due to a decrease in the training time. I trained the CNN models with only 100 iterations. Furthermore, the loss value of the training decreased quite rapidly becoming very close to zero at just iteration 40 to iteration 50. In the experiment, I found that the best and robust spatial features were extracted using ResNet50 architecture. I then combined the ResNet50 with the Conv1D-LSTM, called ResNet50+Conv1D-LSTM. The experimental result showed that the ResNet50+Conv1D-LSTM network significantly outperformed other CNNs on the ETH food-101 dataset.

Moreover, I also experimented with data augmentation techniques, including rotation, width shift, height shift, horizontal flip, shear, and zoom. The data augmentation techniques consistently achieved better performance.

In Chapter 4, I improved the efficiency performance for food image recognition by investigating an adaptive feature fusion network (ASTFF-Net). With the ASTFF-Net, I obtained robust features generated from CNN models at different layers. Here, I proposed several ASTFF-Net models that were a combination between state-of-the-art CNN models and the LSTM network with improved the performance of the food image recognition system. Motivated by the Conv1D-LSTM network described in Chapter 3, our ASTFF-Net was invented to capture the robust deep features on both spatial and temporal features from the variation of the real-world

food images. I first extracted the spatial features using state-of-the-art ResNet50 architecture. Second, the temporal features were extracted using the LSTM network. Third, the deep features extracted from CNN and LSTM networks were mapped to a similar resolution before concatenating. Finally, I attached extra layers to prevent overfitting before sending the deep adaptive features to the softmax function. The proposed ASTFF-Net achieved the best performances and outperformed other methods on Food11, UEC Food-100, UEC Food-256, and ETH Food-101.

In this dissertation, three robust approaches to improve the accuracy of food image recognition are proposed, including the modified MobileNet architecture, the Conv1D-LSTM network, and the ASTFF-Net.

5.1 Answers to The Research Questions

According to the research questions (RQ) in Chapter 1, I explain the improvement of the food image recognition systems based on real-world food images with three solutions. In this section, I briefly answer each research question.

RQ1: Training the model with deep learning methods such as convolutional neural network (CNN) typically requires a large amount of training data to create an effective model (Russakovsky et al., 2015). The benchmark food image datasets, such as the ETH food-101, contain 101,000 real-world food images (Bossard & Gool, 2014). Indeed, the CNN architectures spent expensive training time to create the effective CNN model. Is it possible to decrease the size of the training data although still provide the same performance of the recognition?

To find out the answer to RQ1, I will focus on modifying a state-of-the-art lightweight CNN model. The hyperparameters and computational layers of the CNN model are also considered. Moreover, I will consider the data augmentation techniques that benefit learning to build an effective CNN model from distinctive food images. Will these methods encourage improving the performance of food image recognition systems?

To answer RQ1, I first focused on the publicly available dataset for food image recognition, namely ETH food-101. It has 101,000 real-world food images of

101 categories and contains 1,000 images in each category. Second, to reduce the computation time, I then selected the state-of-the-art lightweight MobileNetV1 architecture. Further, I modified the MobileNetV1 by eliminated the average pooling layer and the fully connected layer (FC). Hence, the global average pooling layer (GAP), the batch normalization layer (BN), rectified linear unit (ReLU) activation function, and dropout layers were attached instead. Third, I decided to use the data augmentation techniques consisting of rescaling, rotation, width shift, height shift, horizontal flip, shear, zoom, and random cropping. The accuracy of results increased approximately 5% when training the modified MobileNet model with applied data augmentation techniques. Both modified MobileNet and the data augmentation techniques are proposed to prevent overfitting. Finally, I experimented with the size of the training data. Consequently, I can reduce the training size from 80,800 images to only 40,400 images but still obtain high performance compared to other research.

Our modified MobileNet architecture makes a model relatively small, requires less computation time, and achieves high performance on the food image recognition systems.

RQ2: In computer vision, hand-crafted feature techniques are presented to extract the specific information existing in the image. Indeed, it mainly focuses on extracting local features. The well-known hand-crafted feature techniques, include local binary pattern (LBP) (Ojala et al., 1994), histogram of oriented gradient (HOG) (Dalal & Triggs, 2005), scale-invariant feature transform (SIFT) (Lowe, 2004), and speeded up robust features (SURF) (Bay et al., 2008). Nowadays, the CNN technique is a competent procedure that includes feature extraction and recognition. For the feature extraction, the CNN can extract robust special features, including low-level and high-level features, called the deep feature extraction method (Y. Chen et al., 2016; Paul et al., 2016). Is it a potential approach to manipulate real-world food images that also contain many categories of object other than the food subject? If possible, I will then be interested in using state-of-the-art CNN architecture to extract the deep features and enhance the food image recognition system.

RQ2 is mainly focused on deep feature extraction techniques instead of hand-crafted feature extraction techniques. To answer this research question, I first proposed extracting the spatial features from the well-known CNN, including VGGNet, ResNet, DenseNet, and MobileNet. In the case of the food images, I found that the ResNet architecture provided robust features. Second, to extract the spatial features, I transferred them into the convolutional 1D (Conv1D) followed by long short-term memory (LSTM) network. This method is called Conv1D-LSTM. Finally, the temporal features that were extracted using the Conv1D-LSTM network were then sent to the global average pooling layer (GAP) to minimize the size of the feature before classifying using the softmax function. Furthermore, while training the model, I added six data augmentation techniques; rotation, width shift, height shift, horizontal flip, shear, and zoom. With the data augmentation techniques, the method still provides higher performance. However, in our case, it gained up only 1%.

To confirm that our method performed well on the food image dataset, I evaluated my proposed Conv1D-LSTM network on the ETH food-101 dataset and compared the result with other research. I found that the ResNet50 architecture when combined with the Conv1D-LSTM network, called ResNet50+Conv1D-LSTM, outperformed all other methods on the ETH food-101 dataset.

I also experimented with a deep feature extraction technique base on Conv1D and LSTM Network. The state-of-the-art ResNet architecture was invented to extract the robust features from food images and was employed as the input data for the Conv1D combined with a long short-term memory (LSTM) network. Then, the output of the LSTM was assigned to the global average pooling layer before passing to the softmax function to create a probability distribution. The experimental results showed that using the CNN method to extract special features from food images and through them to the long short-term memory (LSTM) algorithm to extracted temporal features, increases the efficiency of food image recognition.

RQ3: The deep feature extraction method always provides robust features and guarantees high accuracy performance on the real-world food image dataset

(Phiphitphatphaisit & Surinta, 2021). Is there any approach will succeed using the deep feature extraction method using Conv1D and LSTM networks?

To answer the last question in RQ3, I proposed an adaptive feature fusion network (ASTFF-Net) to deal with the real-world food image datasets. In our network, the ASTFF-Net combined three main networks; CNN, Conv1D, and LSTM. First, the state-of-the-art CNN architecture was proposed to extract the spatial features from the food images. Second, I assigned the spatial features to the LSTM network to generate the temporal features. Third, I combined the deep features extracted from the CNN and LSTM networks using the concatenate operation, called the adaptive feature fusion method. I also created extra layers that were used to overcome overfitting. Eventually, the proposed ASTFF-Net obtained the best accuracy on four food image datasets; Food11, UEC Food-100, UEC Food-256, and ETH Food-101.

5.2 Future Work

In this dissertation, I presented novel deep feature extraction techniques to improve the performance of food image recognition based on real-world food images. However, there is still a need to create new deep feature extraction methods or for optimizing the current methods.

In the case of several training data, more computation time is used to create robust CNN models. I then focused on reducing the training data size by applying the instance selection method (Branikas, Papastergiou, Zacharaki, & Megalooikonomou, 2019). This method could be selected the most relevant instance to represent as the training data.

In real-world food image datasets, food image datasets contain food images taken from various orientations. There are always other objects in the food images. performance of the food image recognition system will be improved when I can segment and learn only at the exact food location. In this case, if I visualize the class activation mapping of CNN models, I can understand where the CNN models localize relevant image regions. So, I can implement the technique to select only the particular

food location. I plan to work on the instance segmentation technique (Hafiz & Bhat, 2020).

REFERENCES

Abas, M. A. H., Ismail, N., Yassin, A., & Taib, M. (2018). VGG16 for Plant Image Classification with Transfer Learning and Data Augmentation. International Journal of Engineering and Technology, 7, 90–94. DOI: https://dx.doi.org/10.14419/ijet.v7i4.11.20781

Aditi, Nagda, M. K., & Poovammal, E. (2019). Image Classification Using a Hybrid LSTM-CNN Deep Neural Network. International Journal of Engineering and Advanced Technology, 8(6), 1342–1348. DOI: https://doi.org/10.35940/ijeat.F8602.088619

Aggarwal, U., Popescu, A., & Hudelot, C. (2020) In the IEEE Winter Conference on Applications of Computer Vision (WACV), 1428-1437. DOI: https://doi.org/10.1109/WACV45572.2020.9093475.

Aguilar, E., Bolaños, M., & Radeva, P. (2017a). Exploring Food Detection Using CNNs. In Computer Aided Systems Theory (EUROCAST), 339-347. DOI: https://doi.org/10.1007/978-3-319-74727-9_40

Aguilar, E., Bolaños, M., & Radeva, P. (2017b). Food Recognition using Fusion of Classifiers Based on CNNs. In: Image Analysis and Processing (ICIAP), 1-12. DOI: https://doi.org/10.1007/978-3-319-68548-9_20

Al-Abed, A.-A. A. A. (2021). Obesity-Linked Diseases (Comorbidities). In Obesity and its Impact on Health. Singapore: Springer. DOI: https://doi.org/10.1007/978-981-33-6408-0_8

Altman, N. S. (1992). An Introduction to Kernel and Nearest Neighbor Nonparametric Regression. The American Statistician, 46(3 (Aug., 1992)), 175–185. DOI: https://doi.org/10.2307/2685209

Anthimopoulos, M., Gianola, L., Scarnato, L., Diem, P., & Mougiakkou, S. (2014). A Food Recognition System for Diabetic Patients Based on An Optimized Bag of Features Model. IEEE Journal of Biomedical and Health Informatics, 18(4), 1261–1271. DOI: https://doi.org/10.1109/JBHI.2014.2308928

Attokaren, D. J., Fernandes, I. G., Sriram, A., Murthy, Y. V. S., & Koolagudi, S. G. (2017). Food Classification from Images Using Convolutional Neural Networks. In The 2017 IEEE Region 10 Conference (TENCON), 2801–2806. DOI: https://doi.org/10.1109/TENCON.2017.8228338

Bay, H., Ess, A., Tuytelaars, T., & Van Gool, L. (2008). Speeded-Up Robust Features (SURF). Computer Vision and Image Understanding, 110(3), 346–359. DOI: https://doi.org/10.1016/j.cviu.2007.09.014

Bolanos, M., & Radeva, P. (2016). Simultaneous Food Localization and Recognition. In 23rd International Conference on Pattern Recognition (ICPR), 3140–3145. DOI: https://doi.org/10.1109/ICPR.2016.7900117

Bossard, L., & Gool, L. Van. (2014). Food-101 – Mining Discriminative Components with Random Forests. In the European Conference on Computer Vision (ECCV), 446-461. DOI: https://doi.org/10.1007/978-3-319-10599-4_29

Boureau, Y.-L., Ponce, J., & LeCun, Y. (2010). A Theoretical Analysis of Feature Pooling in Visual Recognition. In the 27th International Conference on International Conference on Machine Learning (ICML), 111-118.

Branikas, E., Papastergiou, T., Zacharaki, E., & Megalooikonomou, V. (2019). Instance Selection Techniques for Multiple Instance Classification. In the 10th International Conference on Information, Intelligence, Systems and Applications (IISA), 1–7. DOI: https://doi.org/ 10.1109/IISA.2019.8900679

Burke, L., Wang, J., & Sevick, M. (2011). Self-monitoring in Weight Loss: A Systematic Review of The Literature. Journal of the American Dietetic Association, 111(1), 92–102. DOI: https://doi.org/10.1016/j.jada.2010.10.008

Butryn, M., Phelan, S., Hill, J. O., & Wing, R. (2007). Consistent Self-monitoring of Weight: A Key Component of Successful Weight Loss Maintenance. Obesity, 15(12),3091-3096. DOI: https://doi.org/10.1038/oby.2007.368

Chaput, J., Klingenberg, L., Astrup, A., & Sjödin, A. (2011). Modern Sedentary Activities Promote Overconsumption of Food in Our Current Obesogenic Environment. Obesity Reviews, 12(5), 12-20. DOI: https://doi.org/10.1111/j.1467-789X.2010.00772.x

Chen, J., Wang, Y., Wu, Y., & Cai, C. (2017). An Ensemble of Convolutional Neural Networks for Image Classification Based on LSTM. In International Conference on Green Informatics (ICGI), 217–222. DOI: https://doi.org/10.1109/ICGI.2017.36

Chen, X., Zhu, Y., Zhou, H., Diao, L., & Wang, D. (2017). ChineseFoodNet: A Large-Scale Image Dataset for Chinese Food Recognition. arXiv:1705.02743v3 [cs.CV], 1–8.

Chen, Y., Jiang, H., Li, C., Jia, X., & Ghamisi, P. (2016). Deep Feature Extraction and Classification of Hyperspectral Images Based on Convolutional Neural Networks. IEEE Transactions on Geoscience and Remote Sensing, 54, 6232–6251. DOI: https://doi.org/10.1109/TGRS.2016.2584107

Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., … Ng, A. (2011). Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning. In 2011 International Conference on Document Analysis and Recognition, 440–445. DOI: https://doi.org/10.1109/ICDAR.2011.95

Cortes, C., & Vapnik, V. (1995). Support-Vector Networks. Machine Learning, 297(20), 273–297. DOI: https://doi.org/10.1111/j.1747-0285.2009.00840.x

Csurka, G. (2004). Visual Categorization with Bags of Keypoints. In Workshop on Statistical Learning in Computer Vision (ECCV), 1–22. DOI: https://doi.org/10.1.1.72.604

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), 1, 886–893. DOI: https://doi.org/10.1109/CVPR.2005.177

Dong, T., Sun, Y., & Zhang, F. (2019). A Diet Control and Fitness Assistant Application using Deep Learning-Based Image Classification. In the 8th International Conference on Natural Language Processing (NLP), 63-98. DOI: https://doi.org/ 10.5121/csit.2019.91207

Ege, T., & Yanai, K. (2017a). Estimating Food Calories for Multiple-Dish Food Photos. In the 4th IAPR Asian Conference on Pattern Recognition (ACPR), 646–651. DOI: https://doi.org/ 10.1109/ACPR.2017.145.

Ege, T., & Yanai, K. (2017b). Image-Based Food Calorie Estimation Using Knowledge on Food Categories, Ingredients and Cooking Directions. In Proceedings of the on Thematic Workshops of ACM Multimedia 2017, 367–375. DOI: https://doi.org/10.1145/3126686.3126742

Farooq, M., & Sazonov, E. (2017). Feature Extraction using Deep Learning for Food Type Recognition. In International Conference on Bioinformatics and Biomedical Engineering (IWBBIO), 464–472. DOI: https://doi.org/10.1007/978-3-319-56148-6_41

Fatehah, A. A., Poh, B. K., Shanita, S. N., & Wong, J. E. (2018). Feasibility of Reviewing Digital Food Images for Dietary Assessment among Nutrition Professionals. Nutrients, 10(8). DOI: https://doi.org/10.3390/nu10080984

Fayyaz, S., & Ayaz, Y. (2019). CNN and Traditional Classifiers Performance for Sign Language Recognition. In Proceedings of the 3rd International Conference on Machine Learning and Soft Computing (ICMLSC 2019). Association for Computing Machinery, New York, NY, USA, 192–196. DOI: https://doi.org/10.1145/3310986.3311011

Giovany, S., Putra, A., Hariawan, A. S., & Wulandhari, L. A. (2017). Machine Learning and SIFT Approach for Indonesian Food Image Recognition. In 2nd International Conference on Computer Science and Computational Intelligence (ICCSCI), 116, 612–620. DOI: https://doi.org/10.1016/j.procs.2017.10.020

Habiba, S. U., Islam, M. F., & Ahsan, S. M. M. (2019). Bangladeshi Plant Recognition using Deep Learning based Leaf Classification. In 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2), 1–4. DOI: https://doi.org/10.1109/IC4ME247184.2019.9036515.

Hafiz, A. M., & Bhat, G. M. (2020). A Survey on Instance Segmentation: State of The Art. International Journal of Multimedia Information Retrieval, 9, 171–189. DOI: https://doi.org/10.1007/s13735-020-00195-x

Haque, M. A., Verma, A., Alex, J. S. R., & Venkatesan, N. (2020). Experimental Evaluation of CNN Architecture for Speech Recognition. In First International Conference on Sustainable Technologies for Computational Intelligence, 507–514. DOI: https://doi.org/10.1007/978-981-15-0029-9_40

Hassannejad, H., Matrella, G., Ciampolini, P., De Munari, I., Mordonini, M., & Cagnoni, S. (2016). Food Image Recognition Using Very Deep Convolutional Networks. In the 2nd International Workshop on Multimedia Assisted Dietary Management, 41–49. DOI: https://doi.org/10.1145/2986035.2986042

He, J., Mao, R., Shao, Z., Wright, J. L., Kerr, D., Boushey, C., & Zhu, F. (2021). An End-to-End Food Image Analysis System. arXiv:2102.00645v1 [cs.CV], 1-5.

He, K., Zhang, X., Ren, S., & J., S. (2016). Deep Residual Learning for Image Recognition. In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778. DOI: https://doi.org/10.1109/CVPR.2016.90.

Hinton, G. E. (2009). Deep belief networks. Scholarpedia, 4, 5947.

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. Neural Computation, 9(8), 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., … Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. In Computer Vision and Pattern Recognition,1,1-9.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2261–2269. DOI: https://doi.org/10.1109/CVPR.2017.243

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv:1502.03167v3, 1-11.

Jain, S., Gupta, R., & Moghe, A. A. (2018). Stock Price Prediction on Daily Stock Data using Deep Neural Networks. 2018 International Conference on Advanced Computation and Telecommunication (ICACAT), 1–13.

Jiang, L., Qiu, B., Liu, X., Huang, C., & Lin, K. (2020). DeepFood: Food Image Analysis and Dietary Assessment via Deep Model. IEEE Access, 8, 47477–47489. DOI:https://doi.org/10.1109/ACCESS.2020.2973625.

Kawano, Y., & Yanai, K. (2014a). Food image recognition with deep convolutional features. In the ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication (UbiComp), 589–593. DOI: https://doi.org/10.1145/2638728.2641339

Kawano, Y., & Yanai, K. (2014b). FoodCam-256: A Large-scale Real-time Mobile Food Recognition System employing High-Dimensional Features and Compression of Classifier Weights. In Proceedings of the 22nd ACM international conference on Multimedia (MM '14). Association for Computing

Machinery, New York, NY, USA, 761–762. DOI: https://doi.org/10.1145/2647868.2654869

Kawano, Y., & Yanai, K. (2015). Automatic Expansion of a Food Image Dataset Leveraging Existing Categories with Domain Adaptation. In Computer Vision - ECCV 2014 Workshops, 3–17. DOI: https://doi.org/10.1007/978-3-319-16199-0_1

Kesav, N., & Jibukumar, M. G. (2021). Complexity Reduced Bi-channel CNN for Image Classification. In Machine Learning for Predictive Analysis, 119–131.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Communications of the ACM, 60(6), 84–90. DOI: https://doi.org/10.1145/3065386

Kumar, V., Namboodiri, A., & Jawahar, C. V. (2020). Region Pooling with Adaptive Feature Fusion for End-to-End Person Recognition. In IEEE Winter Conference on Applications of Computer Vision (WACV), 2122–2131. DOI: https://doi.org/10.1109/WACV45572.2020.9093631

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. In Proceedings of the IEEE, 86(11), 2278-2324, DOI: https://doi.org/10.1109/5.726791.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. In Proceedings of the IEEE, 86(11), 2278-2324, DOI: https://doi.org/10.1109/5.726791.

Li, X., Li, W., Ren, D., Zhang, H., Wang, M., & Zuo, W. (2020). Enhanced Blind Face Restoration with Multi-Exemplar Images and Adaptive Spatial Feature Fusion. In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1, 2703–2712. DOI: https://doi.org/10.1109/CVPR42600.2020.00278

Lin, M., Chen, Q., & Yan, S. (2014). Network In Network. In arXiv:1312.4400v3, 1-10.

Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., & Ma, Y. (2016). DeepFood: Deep Learning-Based Food Image Recognition for Computer-Aided Dietary Assessment. In Inclusive Smart Cities and Digital Health (ICOST), 37-48. DOI: https://doi.org/10.1007/978-3-319-39601-9_4

Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., Yunsheng, M., … Hou, P. (2018). 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. DOI: https://doi.org/10.1109/TSC.2017.2662008

Liu, X., Chi, M., Zhang, Y., & Qin, Y. (2018). Classifying High Resolution Remote Sensing Images by Fine-Tuned VGG Deep Networks. In IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, 7137–7140. DOI: https://doi.org/10.3390/rs9050498

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60, 91–110.

Martinel, N., Foresti, G. L., & Micheloni, C. (2018). Wide-Slice Residual Networks for Food Recognition. In IEEE Winter Conference on Applications of Computer Vision (WACV) , 567–576. DOI: https://doi.org/10.1109/WACV.2018.00068

Martinel, N., Piciarelli, C., & Micheloni, C. (2016). A Supervised Extreme Learning Committee for Food Recognition. Computer Vision and Image Understanding, 148, 67–86. DOI: https://doi.org/10.1016/j.cviu.2016.01.012

Matsuda, Y., & Yanai, K. (2012). Multiple-Food Recognition Considering Co-Occurrence Employing Manifold Ranking. In the 21st International Conference on Pattern Recognition (ICPR), 2017–2020.

McAllister, P., Zheng, H., Bond, R., & Moorhead, A. (2018). Combining Deep Residual Neural Network Features with Supervised Machine Learning Algorithms to Classify Diverse Food Image Datasets. Computers in Biology and Medicine, 95(May 2017), 217–233. DOI: https://doi.org/10.1016/j.compbiomed.2018.02.008

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 MultiMedia Modeling, 129–141.

Must, A., Spadano, J., Coakley, E. H., Field, A. E., Colditz, G., & Dietz, W. H. (1999). The Disease Burden Associated with Overweight and Obesity. JAMA, 282(16), 1523–1529. DOI: https://doi.org/10.1001/jama.282.16.1523

Myers, A., Johnston, N., Rathod, V., Korattikara, A., Gorban, A., Silberman, N., … Murphy, K. (2015). Im2Calories: Towards an Automated Mobile Vision Food

Diary. In 2015 IEEE International Conference on Computer Vision (ICCV), 1233–1241. DOI: https://doi.org/10.1109/ICCV.2015.146

Nair, V., & Hinton, G. E. (2010). Rectified Linear Units Improve Restricted Boltzmann Machines. In the 27th International Conference on International Conference on Machine Learning (ICML), 807-814.

Nanni, L., Ghidoni, S., & Brahnam, S. (2017). Handcrafted vs. Non-Handcrafted Features for Computer Vision Classification. Pattern Recognition, 71, 158–172.

Ng, Y. Sen, Xue, W., Wang, W., & Qi, P. (2019). Convolutional Neural Networks for Food Image Recognition: An Experimental Study. In the 5th International Workshop on Multimedia Assisted Dietary Management (MADiMa), 33–41. DOI: https://doi.org/10.1145/3347448.3357168

Nguyen, B. T., Dang-Nguyen, D.-T., Tien, D. X., Phat, T. Van, & Gurrin, C. (2018). A Deep Learning based Food Recognition System for Lifelog Images. In 7th International Conference on Pattern Recognition Applications and Methods (ICPRAM), 657-664. DOI: https://doi.org/10.5220/0006749006570664

Nguyen, D. T., Zong, Z., Ogunbona, P. O., Probst, Y., & Li, W. (2014). Food Image Classification Using Local Appearance and Global Structural Information. Neurocomputing, 140, 242–251. DOI: https://doi.org/10.1016/j.neucom.2014.03.017

Nordin, M. J., Xin, O. W., & Aziz, N. (2019). Food Image Recognition for Price Calculation using Convolutional Neural Network. In the 3rd International Conference on Digital Signal Processing (ICDSP), 80–85. DOI: https://doi.org/10.1145/3316551.3316557

Nwankpa, C., Ijomah, W., Gachagan, A., & Marshall, S. (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning. In 2nd International Conference on Computational Sciences and Technologies, 124-133.

Ojala, T., Pietikainen, M., & Harwood, D. (1994). Performance Evaluation of Texture Measures with Classification Based on Kullback Discrimination of Distributions. In 12th International Conference on Pattern Recognition (ICPR), 1, 582–585. DOI: https://doi.org/10.1109/ICPR.1994.576366

Okafor, E., Schomaker, L., & Wiering, M. (2018). An Analysis of Rotation Matrix and Colour Constancy Data Augmentation in Classifying Images of Animals. J. Information Telecommunication, 2, 465–491.

Pandey, P., Deepthi, A., Mandal, B., & Puhan, N. B. (2017). FoodNet: Recognizing Foods Using Ensemble of Deep Networks. IEEE Signal Processing Letters, 24(12), 1758–1762. DOI: https://doi.org/10.1109/LSP.2017.2758862

Park, S.-J., Palvanov, A., Lee, C.-H., Jeong, N., Cho, Y.-I., & Lee, H.-J. (2019). The Development of Food Image Detection and Recognition Model of Korean Food for Mobile Dietary Management. Nutrition Research and Practice, 13, 521–528.

Paul, R., Hawkins, S. H., Balagurunathan, Y., Schabath, M., Gillies, R., Hall, L., & Goldgof, D. (2016). Deep Feature Transfer Learning in Combination with Traditional Features Predicts Survival Among Patients with Lung Adenocarcinoma. Tomography, 2, 388–395.

Pawara, P., Okafor, E., & Schomaker, L. (2017). Data Augmentation for Plant Classification. Advanced Concepts for Intelligent Vision Systems (ACIVS), 615-626. DOI: https://doi.org/10.1007/978-3-319-70353-4_52

Pearline, S. A., Vajravelu, S. K., & Harini, S. (2019). A Study on Plant Recognition Using Conventional Image Processing and Deep Learning Approaches. J. Intell. Fuzzy Syst., 36, 1997–2004. DOI: https://doi.org/10.3233/JIFS-169911

Phiphiphatphaisit, S., & Surinta, O. (2020). Food Image Classification with Improved MobileNet Architecture and Data Augmentation. In the 3rd International Conference on Information Science and Systems (ICISS), 51–56. DOI: https://doi.org/10.1145/3388176.3388179

Phiphitphatphaisit, S., & Surinta, O. (2021). Deep Feature Extraction Technique Based on Conv1D and LSTM Network for Food Image Recognition. Engineering and Applied Science Research, 48(5), 581–592. DOI: https://doi.org/10.14456/easr.2021.60

Ragusa, F., Tomaselli, V., Furnari, A., Battiato, S., & Farinella, G. (2016). Food vs Non-Food Classification. In the 2nd International Workshop on Multimedia Assisted Dietary Management (MADiMa), 77–81. DOI: https://doi.org/10.1145/2986035.2986041.

Ritchie, H., & Roser, M. (2017). Obesity. Retrieved September 9, 2021, from https://ourworldindata.org/obesity

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 115(3), 211–252. DOI: https://doi.org/10.1007/s11263-015-0816-y

Sahoo, D., Hao, W., Ke, S., Wu, X., Le, H., Achananuparp, P., Lim, E., & Hoi, S.C. (2019). FoodAI: Food Image Recognition via Deep Learning for Smart Food Logging. In the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2260–2268. DOI: https://doi.org/10.1145/3292500.3330734.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. In IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4510–4520. DOI: https://doi.org/10.1109/CVPR.2018.00474

Sasano, S., Han, X. H., & Chen, Y. W. (2017). Food Recognition by Combined Bags of Color Features and Texture Features. In Proceedings - 2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics, CISP-BMEI 2016, 815–819. https://doi.org/10.1109/CISP-BMEI.2016.7852822

Sengur, A., Akbulut, Y., & Budak, U. (2019). Food Image Classification with Deep Features. In International Artificial Intelligence and Data Processing Symposium (IDAP), 1–6. DOI: https://doi.org/10.1109/IDAP.2019.8875946

Shahzadi, I., Tang, T., Meriadeau, F., & Quyyum, A. (2018). CNN-LSTM: Cascaded Framework for Brain Tumour Classification. 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), 633–637.

Shrestha, A., & Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. IEEE Access, 7, 53040–53065. DOI: https://doi.org/10.1109/ACCESS.2019.2912200

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. In arXiv:1409.1556v6, 1-14.

Singla, A., Yuan, L., & Ebrahimi, T. (2016). Food/Non-food Image Classification and Food Categorization using Pre-Trained GoogLeNet Model. In Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, 3–11. DOI: https://doi.org/10.1145/2986035.2986039.

Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15, 1929–1958.

Takahashi, R., Matsubara, T., & Uehara, K. (2020). Data Augmentation Using Random Image Cropping and Patching for Deep CNNs. IEEE Transactions on Circuits and Systems for Video Technology, 30(9), 2917–2931. DOI: https://doi.org/10.1109/TCSVT.2019.2935128

Tanno, R., Okamoto, K., & Yanai, K. (2016). DeepFoodCam: A DCNN-based Real-Time Mobile Food Recognition System. MADiMa 2016 - Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management, Co-Located with ACM Multimedia 2016, 89. DOI: https://doi.org/10.1145/2986035.2986044

Tasci, E. (2020). Voting Combinations-Based Ensemble of Fine-Tuned Convolutional Neural Networks for Food Image Recognition. Multimedia Tools and Applications, 79(41–42), 30397–30418. DOI: https://doi.org/10.1007/s11042-020-09486-1

Vijayakumar, D. S., & Sneha, M. (2021). Low Cost Covid-19 Preliminary Diagnosis Utilizing Cough Samples and Keenly Intellective Deep Learning Approaches. Alexandria Engineering Journal, 60(1), 549–557. DOI: https://doi.org/10.1016/j.aej.2020.09.032

World Health Organization. (2018). Obesity and Overweight. Retrieved September 9, 2021, from http://www.who.int/mediacentre/factsheets/fs311/en/

World Population Review. (2021). Obesity Rates By Country 2021. Retrieved September 9, 2021, from https://worldpopulationreview.com/country-rankings/obesity-rates-by-country

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional Neural Networks: an Overview and Application in Radiology. Insights into Imaging, 9(4), 611–629. DOI: https://doi.org/10.1007/s13244-018-0639-9

Yan, J., Qi, Y., & Rao, Q. (2018). Detecting Malware with an Ensemble Method Based on Deep Neural Network. Security and Communication Networks, 1-16. DOI: https://doi.org/10.1155/2018/7247095

Yanai, K., & Kawano, Y. (2015). Food Image Recognition using Deep Convolutional Network with Pre-training and Fine-Tuning. In IEEE International Conference on Multimedia & Expo Workshops (ICMEW), 1–6. DOI: https://doi.org/10.1109/ICMEW.2015.7169816

Yang, M., Zhang, J., Yang, Y., & Wen, C. (2018). Modelling Spatial Correlations by Using Deep CNN and LSTM for Texture Image Classification. In IEEE 27th International Symposium on Industrial Electronics (ISIE), 759–764.

Yunus, R., Arif, O., Afzal, H., Amjad, M. F., Abbas, H., Bokhari, H. N., … Nawaz, R. (2019). A Framework to Estimate the Nutritional Value of Food in Real Time Using Deep Learning Techniques. IEEE Access, 7, 2643–2652. https://doi.org/10.1109/ACCESS.2018.2879117

Zainudin, Z., Shamsuddin, S., & Hasan, S. (2019). Convolutional Neural Network Long Short-Term Memory (CNN + LSTM) for Histopathology Cancer Image Classification. In Machine Intelligence and Signal Processing (MISP), 235-245. DOI: https://doi.org/10.1007/978-981-15-1366-4_19

Zhao, S., Xu, T., Wu, X. J., & Zhu, X. F. (2021). Adaptive Feature Fusion for Visual Object Tracking. Pattern Recognition, 111, 107679. DOI: https://doi.org/10.1016/j.patcog.2020.107679

Zheng, J., Zou, L., & Wang, Z. (2018). Mid-level Deep Food Part Mining for Food Image Recognition. IET Computer Vision, 12, 298–304. DOI: https://doi.org/10.1049/iet-cvi.2016.0335

BIOGRAPHY

NAME	Sirawan Phiphitphatphaisit
DATE OF BIRTH	16 April 1976
PLACE OF BIRTH	Bankkok
ADDRESS	255/7 Village No. 5, Phra Lap Subdistrict, Mueang District, Khon Kaen Province
POSITION	Lecturer
PLACE OF WORK	Rajamangala University of Technology Khon Kaen Campus
EDUCATION	1998 Bachelor of Science (B.Sc.) Computer Science, Nakhon Ratchasima Rajabhat Institute 2003 Master of Science (M.Sc.) Information Technology, King Mongkut's Institute of Technology North Bangkok 2022 Doctor of Philosophy (Ph.D.) Information Technology, Mahasarakham University
Research output	1. Phiphiphatphaisit, S., & Surinta, O. (2020). Food Image Classification with Improved MobileNet Architecture and Data Augmentation. Proceedings of the 2020 The 3rd International Conference on Information Science and System. 2. Chompookham, T., Gonwirat, S., Lata, S., Phiphiphatphaisit, S., & Surinta, O. (2020). Plant Leaf Image Recognition using Multiple-grid Based Local Descriptor and Dimensionality Reduction Approach. Proceedings of the 2020 The 3rd International Conference on Information Science and System. 3. Phiphitphatphaisit, S., & Surinta, O. (2021). Deep feature extraction technique based on Conv1D and LSTM network for food image recognition. Engineering and Applied Science Research, 48(5), pages 581-592.