Automatic Vehicle Detection and Classification System using Advanced Deep Learning Architectures

Narong Boonsirisumpun

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

June 2023

ระบบตรวจจับและจำแนกยานพาหนะแบบอัตโนมัติด้วยสถาปัตยกรรมการเรียนรู้เชิงลึกขั้นสูง

วิทยานิพนธ์

ของ

ณรงค์ บุญสิริสัมพันธ์

เสนอต่อมหาวิทยาลัยมหาสารคาม เพื่อเป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาปรัชญาดุษฎีบัณฑิต สาขาวิชาเทคโนโลยีสารสนเทศ

มิถุนายน 2566

Automatic Vehicle Detection and Classification System using Advanced Deep Learning Architectures

Narong Boonsirisumpun

A Thesis Submitted in Partial Fulfillment of Requirements

for Doctor of Philosophy (Information Technology)

June 2023

The examining committee has unanimously approved this Thesis, submitted by Mr. Narong Boonsirisumpun , 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. Somnuk Puangpronpitag , 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

....

(Assoc. Prof. Jantima Polpinij , Ph.D.) (Assoc. Prof. Krit Chaimoon , Ph.D.)

Dean of The Faculty of Informatics Dean of Graduate School

TITLE	Automatic Vehicle Detection and Classification System using Advanced Deep Learning Architectures		
AUTHOR	Narong Boonsirisumpun		
ADVISORS	Assistant Professor Olarik Surinta , Ph.D.		
DEGREE	Doctor of Philosophy	**MAJOR**	Information Technology
UNIVERSITY	Mahasarakham University	**YEAR**	2023

ABSTRACT

This dissertation aims to develop an automatic vehicle detection and classification system that leverages advanced deep learning architectures. The system comprises three parts. The first part employs advanced convolutional neural networks (CNN) to classify images of five vehicle types in Thailand. By comparing nine different CNN models with data augmentation, we found that MobileNets is the best method in terms of accuracy, speed, and size. The second part uses ensemble methods to combine multiple CNN models to recognize vehicle type and make (logo) using a "partial training set" technique. This approach improves accuracy and reduces overall runtime. In the third part, we propose a hybrid structure of a generative adversarial network (GAN) and a CNN for object detection using YOLO technique to recognize Thai license plates. By testing different GAN architectures and YOLO networks, we found that the hybrid of ESRGAN-YOLOv7 outperformed other combinations in terms of accuracy. Overall, this dissertation provides a comprehensive solution to the problem of automatic vehicle detection and classification using the latest deep learning methods, highlighting the importance of using ensemble methods and partial training sets to improve accuracy and reduce runtime. The proposed system has the potential to be utilized in various real-world applications, including video surveillance systems and mobile devices.

Keyword : Vehicle Detection and Classification System, Deep Learning, Thai License Plate Recognition, Convolutional Neural Network, Ensemble Method, Partial Training Set, Data Augmentation, Generative Adversarial Network, Hybrid GAN-YOLO

ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my academic advisor, Assistant Professor Olarik Surinta, for his valuable guidance, constant support, and encouragement throughout the five years of my study. His expertise, insights, and patience have been instrumental in shaping the direction and quality of this research, and I have learned a lot from him not only about deep learning and computer vision but also about research ethics and academic professionalism.

I would also like to thank the members of our research team, Multi-agent Intelligent Simulation Laboratory (MISL), Department of Information Technology, Faculty of Informatics, Mahasarakham University, for the collaboration, feedback, and inspiration. Our diverse backgrounds and perspectives have enriched my understanding of this field and motivated me to explore new ideas and methods.

Furthermore, I would like to extend my sincere appreciation to the Office of Academic Resources and Information Technology, Loei Rajabhat University, for the support in data collection process of three datasets used in this research. Without their cooperation and assistance, this research would not have been possible.

Finally, I would like to thank my family and friends for their love, patience, and understanding during this challenging and rewarding journey. Their constant encouragement and belief in me have been a great source of motivation and comfort.

Narong Boonsirisumpun

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

CHAPTER I

INTRODUCTION

1.1 Introduction

Vehicle detection and classification is a critical problem in the field of computer vision and image processing. The goal of this problem is to automatically identify and classify vehicles in digital images and videos. The task is challenging due to the complex nature of vehicle appearances, variations in lighting and weather conditions, and the presence of occlusions and clutter in the background.

There have been numerous attempts to address this problem using various image processing and machine learning techniques. These techniques involve the detection of vehicle regions in an image or video stream and the classification of the detected regions into specific vehicle types. Many of these techniques are based on feature extraction, segmentation, and classification algorithms, such as deep learning models, support vector machines, and random forests.

Despite the significant progress made in the field of vehicle detection and classification, challenges persist, including the need for robust detection and classification in real-time environments, as well as the ability to extract detailed information about detected vehicles. One of the major challenges in this field is the lack of sufficient training data, which hinders the development of accurate and robust computer vision algorithms for vehicle recognition and tracking. To address this challenge, researchers are exploring new techniques that integrate advanced computer vision algorithms with other data sources, such as radar and lidar. These techniques hold promise for enhancing the performance of vehicle detection and classification in real-world scenarios. The development of more sophisticated vehicle recognition and tracking systems have important applications, such as in the areas of intelligent transportation systems, surveillance, and security.

1.2 Background of the Problem

The increment of traffic problems around the world forced several countries to build a system to assist officers for control and secure the transportation problem. One of those countries is Thailand. Every year an enormous number of cars are sold and increase the complexity for the Thai government to supervise and control transportation. Researchers have tried to solve this problem for several years by adding a street video surveillance system to help police officers capture illegal use of vehicles without human standing on the road as happened in the past. The objective of this system is to detect a vehicle moving in the video image and try to collect the information of that vehicle. Mostly in the system need to identify the vehicle registration data from the license plate character and number in a video image. However, there are several situations that the vehicle license plate cannot be observed such as the obstacle covering up the plate or the image orientation that hiding it from a user perspective (see Figure 1). Forcing the system to look for other details such as vehicle type or vehicle make and model instead of the registration number which takes it to the area of image recognition problem.

Figure 1. Vehicle image without license plate detail.

Another significant challenge faced by the system is the detection of license plates in low-quality images, as shown in Figure 2. This issue can arise due to a suboptimal video image resolution or camera quality. In such cases, the system requires additional algorithms to assist with vehicle information identification or to enhance image quality through the use of "Image Restoration" techniques. To address these challenges, this study proposes a comprehensive solution that leverages three image processing algorithms: image classification, object detection, and image restoration.

Figure 2. Vehicle license plate from a low-quality image.

Several techniques have applied to solve these problems. For example, Artificial Neural Network (Bishop, 1995; Nasrabadi, 2007), Support Vector Machine (SVM) (Vapnik, 1999), Genetic Algorithm (Alsultanny & Aqel, 2003; Pal & Wang, 1996), or K-Nearest Neighbors (Wu et al., 2002; Zhang et al., 2006). However, the progress in another technique called "Convolutional Neural Networks (CNNs)" (LeCun et al., 1998) was proved to be a better solution in this problem. The CNNs is one of the algorithms based on a deep neural network structure which many people prefer to call it "Deep Learning (DL)" (Sermanet et al., 2013; Sharif Razavian et al., 2014). The work on CNNs has been very popular since the introduction of "AlexNet" (Krizhevsky et al., 2017), the first CNNs model that won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012. After that, several models tried to improve and outperform its predecessor such as VGGNet (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy et al., 2015), and MobileNets (Howard et al., 2017).

Those CNNs models are performing well for image classification problem. But for object detection, it needs to add extra features in the model to perform both classification and detection steps in only one network. Several techniques have been succeeded in these tasks. For example, R-CNN (Girshick et al., 2015), Faster R-CNN (Ren et al., 2015), You Only Look Once or YOLO (Redmon et al., 2016) and Single Shot multibox Detector or SSD (Liu et al., 2016). These models can predict the region of an interesting object by choosing the best bounding box area in the image and then identify its type in a single runtime.

The performance of convolutional neural networks (CNNs) in detecting and recognizing objects is highly dependent on the quality of the input image. To address the limitations of low-quality images, the process of "Image Restoration" is employed to adjust image data before submission to the CNNs. Previously, basic methods such as The Inverse filter or The Wiener filter were used for this function. However, with the advent of Generative Adversarial Networks or GANs (Goodfellow et al., 2014), researchers can now recover the quality of low-resolution images or even generate new ones that differ from the source. Notably, recent successful GAN models in image restoration and super-resolution include DCGAN (Radford et al., 2015), SRGAN (Ledig et al., 2017), and DeblurGAN (Kupyn et al., 2018).

1.3 Statement of the Problem

The research aims to address three main challenges in the development of a comprehensive vehicle detection and classification system using deep learning algorithms.

The first challenge is image classification, one of the primary issues is the lack of sufficient training data. This poses a significant hurdle for accurate identification and classification of vehicles in digital images. The deep learning algorithm must accurately identify features such as vehicle type, make, model, or license plate data based on limited training data. Therefore, the accuracy of the classification results is highly dependent on the availability and quality of training data. Insufficient training data can negatively impact the performance of the algorithm and hinder the effectiveness of the system. As such, the evaluation of the algorithm's performance in image classification is critical in assessing the effectiveness of the system and identifying areas for improvement.

The second challenge faced by the system is the detection and localization of vehicles and their license plates in digital images, known as vehicle object detection. Accurate and real-time identification and tracking of vehicles are essential tasks for the system. The efficacy of the deep learning algorithm will be evaluated based on the accuracy and speed of the object detection process.

The final challenge is image restoration, which aims to restore the quality of digital images affected by noise, blur, or low-resolution factors that can adversely impact the overall performance of the vehicle detection and classification system. The effectiveness of the deep learning algorithm in restoring image quality and improving the accuracy of the vehicle detection and classification system will be assessed.

1.4 Purpose of the Study

The primary objective of this dissertation is to develop and evaluate deep learning algorithms for vehicle detection and classification in digital images and videos. The study aims to achieve three main goals:

The first objective is to study and develop a deep learning algorithm that can accurately classify vehicles based on their type, make, or license plate with a limited training data. The algorithm will be trained using a small dataset of vehicle images and evaluated based on its performance in classifying vehicles.

The second objective of this study is to evaluate and analyze the performance of the developed deep learning algorithms using three vehicle image datasets, namely Vehicle Type, Vehicle Make, and Vehicle License Plate. The efficacy of the algorithms in accurately detecting and classifying vehicles based on these datasets will be assessed.

The third objective is to evaluate and analyze the performance of deep learning algorithms for image restoration. The restoration algorithms aim to improve the quality of digital images and videos that have been affected by noise, blur, or other factors, thereby enhancing the performance of the autonomous vehicle detection and classification system. The effectiveness of the restoration algorithm will be evaluated based on its ability to improve the overall performance of the system.

1.5 Research Questions

In this study, the primary focus is on investigating five key research questions related to the use of deep learning in the recognition and classification of vehicles. The first research question pertains to the performance of vehicle type recognition, specifically whether deep learning algorithms can accurately detect and specify the type of vehicle based on its image, such as sedan, hatchback, SUV, pick-up, and van.

The second research question centers around vehicle make recognition and whether deep learning algorithms can accurately identify the brand of a vehicle based on an image alone. Additionally, the study aims to explore the potential of deep learning in performing character recognition on Thai license plates, which is the subject of the third research question.

The fourth research question seeks to determine the most effective model for the vehicle detection and classification problem, taking into account various factors such as accuracy, speed, and ease of implementation. Finally, the fifth research question focuses on the potential for deep learning to improve the quality of low-quality vehicle images through the use of image restoration algorithms.

Overall, this study aims to contribute to the growing body of research on the use of deep learning in computer vision and image recognition, specifically in the context of vehicle detection and classification. By addressing these five key research questions, the study seeks to provide valuable insights into the capabilities and limitations of deep learning algorithms in this domain.

1.6 Scope of Study

In this research, the researcher designs the scope of the study based on each aspect as follows:

1.6.1 Algorithm and Application

In this study, we are focusing on several algorithms based on deep learning techniques which mainly are Convolutional Neural Networks (CNNs), then applications of these algorithms are split into three topics: image classification, object detection, and image restoration.

1.6.2 Evaluation

The evaluations of this research will be measured based on three scopes of measurement as follow:

1. The accuracy of image object classification and detection.
2. The time consuming of each algorithm in the experiment.
3. The performance of the image restoration algorithm and its effects on the accuracy of image object classification and detection.

1.6.3 Dataset

The researcher intends to obtain the vehicle image dataset from the video surveillance system at Loei Rajaphat University, utilizing two cameras as the primary recording devices. The first camera will capture images from the university's front gate, while the second camera will capture images from the rear entrance. The image gathering process aims to collect approximately 1,000 to 4,000 vehicle images, simulating a small dataset scenario.

1.7 Significance of the Study

This study aims to contribute to the field of vehicle recognition and tracking systems in two significant ways:

Firstly, the study addresses the challenge of limited training data by proposing a solution for accurate classification and detection of vehicle type, make, and license plates using deep learning algorithms. The proposed approach is designed to overcome the limitations of training data by leveraging transfer learning techniques and data augmentation to improve the accuracy of the classification results.

Secondly, the significance and benefits of this study lie in its potential to improve the accuracy and efficiency of vehicle recognition and tracking systems in real-world scenarios, such as intelligent transportation systems, surveillance, and security. By overcoming the limitations of image quality and proposing deep learning model for image restoration, this study provides a valuable contribution to the field and opens up new avenues for future research.

1.8 Definition of Terms

1.8.1 Vehicle Type is the description of the vehicle category that helps define the terms for classifying cars or other types of vehicles.

1.8.2 Vehicle Make is the brand of the vehicle, mostly are the name of the company manufacturing the vehicle.

1.8.3 License plate or vehicle registration plate is a metal or plastic plate attached to a vehicle for official identification purposes.

CHAPTER II

LITERATURE REVIEW

This research is theoretical and applied research. The objective is to study and examine the efficiency of various deep learning algorithms to solve the vehicle image classification and detection problem. With has the theories and related research as follows.

2.1 Vehicle Detection System

2.2 Vehicle Type

2.3 Vehicle Make

2.4 Vehicle License Plate

2.5 Deep Learning

2.6 Convolutional Neural Network

2.7 Deep Learning Model for Image Classification

2.8 Deep Learning Model for Image Detection

2.9 Deep Learning Model for Image Restoration

2.10 Relevant Research

2.1 Vehicle Detection System

The increase of the traffic problem around the world have forced several countries to build a system to assist the officer to control and secure the transportation problem, one of those countries is Thailand. Every year a lot of new cars has been selling and make more complexity for the Thai government to supervise the road flow and transportation control. Researchers and officers have solved this problem for several years by adding the video surveillance system in the street (See Figure 3) and help police officer to capture an illegal used of vehicle manually by people without standing on the road like in the past.

Figure 3. Street Video Surveillance in Thailand.

The objective of this system is to detect the vehicle moving in the image and find the detail of that vehicle by extract data from its license plate (See Figure 4). If some vehicles abuse the traffic laws, the video camera will capture the frame and send a report to the police officer. But the problem is this system still not the fully automatic camera system. Some examples are the speed control, a camera cannot detect speed limit by using only its own device but need some additional sensors to help control the camera such as Laser, RADAR, LIDAR, and Infrared (Sawicki, 1993). And after that it needs to do the License Plate Recognition (LPR) step, the standard OCR can recognize the character in the image but it needs to be in the correct orientation with a clearly condition (Mori et al., 1999).

Figure 4. Thai Vehicle Registration License Plate.

2.2 Vehicle Type

Vehicle Type is the description of the vehicle category that helps define the terms for classifying cars or other types of vehicles. In this research, we will focus on five types of mainly personal vehicle in Thailand (See Figure 5):

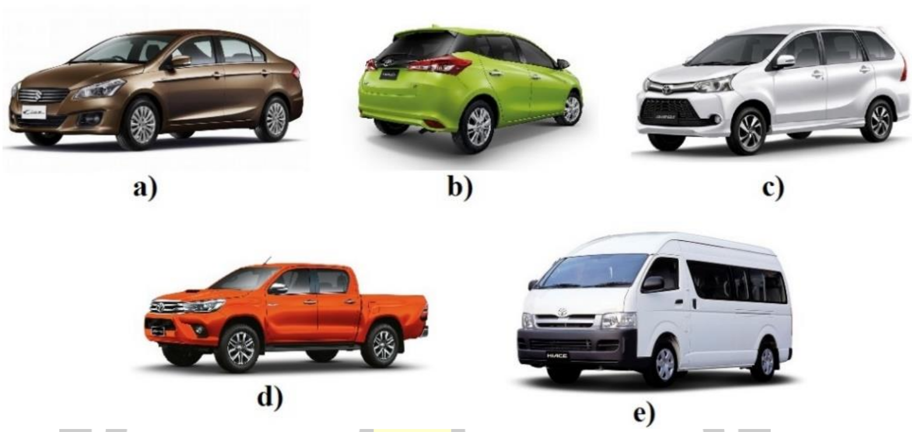


Figure 5. Five types of Vehicles in Thailand.

a) Sedan, b) Hatchback. c) SUV, d) Pick-up, e) Van.

1. Sedan: A passenger car with body of two or four doors and two full-width seats inside.

2. Hatchback: Another passenger car which has a single rear door for storage.

3. Sport utility vehicle (SUV): a larger hatchback which is similar to a station wagon.

4. Pick-up: A small truck with an open body and enclosed cab.

5. Van: A vehicle with three or four passenger seats which can transport more than ten people.

2.3 Vehicle Make

Vehicle Make is the brand of the vehicle, mostly are the name of the company manufacturing the vehicle. Remembering the brand of a vehicle can help officers to reduce the range of data they require to explore before process into another step of traffic control. In Thailand, the automotive industry is mainly dominated by Japanese and American car company (Maikaew, 2018). From figure 6, the biggest market shares of the car industry in Thailand are Toyota (27.2%) followed by Isuzu (18.9%), Honda (14.8%), Mitsubishi (7.9%) and Nissan

(6.9%). Ford is the biggest American car company in Thailand (6.4%) and followed by Chevloret (2.1%).

TOP 12 CAR SALES VOLUME

By market share in 2017 (11 months)

Brand	Sales volume	YoY	Share
• Toyota	208,755	-3.8%	27.2%
• Isuzu	145,108	13.7%	18.9%
• Honda	113,305	16.8%	14.8%
• Mitsubishi	60,824	24.7%	7.9%
• Nissan	52,700	40.3%	6.9%
• Ford	49,428	37.5%	6.4%
• Mazda	45,098	18.5%	5.9%
• Suzuki	21,898	10.1%	2.9%
• Chevrolet	15,983	26.1%	2.1%
• Mercedes-Benz	12,563	20.3%	1.6%
• Hino	10,782	-2.9%	1.4%
• MG	10,085	39.1%	1.3%

Source: Toyota Motor Thailand

Figure 6. The popular car brands in Thailand (2017).

Normally, each company installed its logo on the front and rear middle position of the vehicle. People are easily recognizing the vehicle make by staring at the logo because of its unique design and the familiarity of the people (See Figure 7). This advantage can help a machine to do the same thing. By locating and recognizing the vehicle logo, make it possible for a computer system to classify the vehicle make by analyzing the differences in each logo and figure out how to categorize them.

Figure 7. Example of the vehicle logo.

2.4 Vehicle License Plate

License plate or vehicle registration plate is a metal or plastic plate attached to a vehicle for official identification purposes. Because the registered license plate is the main identification of each vehicle in the traffic control system. If the system can detect and recognize the detail in each license plate, they might be able to access to the full detail of each car in the database (if any) without recognizing the vehicle type or the vehicle brand.

For the five types of Thai vehicle, we focus on this research (Sedan, Hatchback, SUV, Pick-up, and Van), there are only five possible types of the license plate can be used in these types of vehicles (See Figure 8).

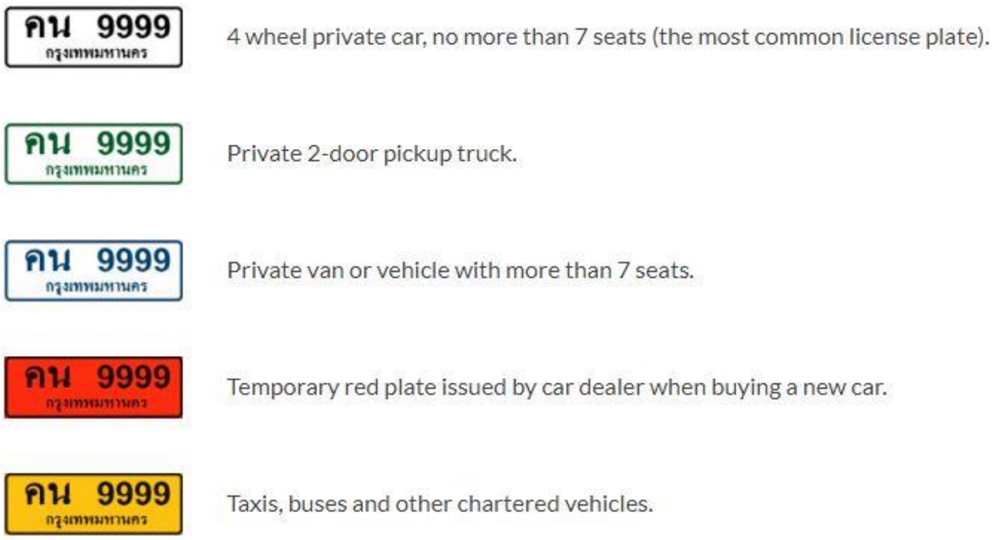


Figure 8. Five types of Thai license plate.

The first one is the black color character license plate, it is the most common license plate for Thai passenger vehicle with usually have less than 7 seats (Sedan, Hatchback, SUV, and 4-door Pick-up). The second type is the green license plate. It is the license plate used only for the private 2-door Pick-up. The third one is the blue license plate, only for the vehicle with more than 7 seats (Commonly a Van). The fourth type uses the black color on the character and the red background color, this type can be used temporary when people buy a new car and wait for the permanent one. The last one uses the yellow background color, this license type only uses by taxi vehicle which can be in several types of vehicles (Sedan, Hatchback, or SUV)

The difficulty of recognizing the license plate is the condition of the plate appearing on the scene. If the plate appears in the image with the straight orientation and the character can be seen clearly, we can easily detect and recognize the information on the plate. But if the license plate shows in the others tilted angle or comes with unclear conditions (For example dirt, shadow, obstacle or plate cover), it is going to be harder for a machine to extract the information from this license plate (See Figure 9).

Figure 9. Example of a license plate with varied condition.

2.5 Deep Learning

Deep Learning (also known as deep structured learning or hierarchical learning) is a part of Machine Learning method based on artificial neural network; the phrase "Deep Learning (DL)" was first used by Rina Dechter in 1986 (Dechter, 1986). The general idea of this technique is the combination of a neural network model that was structured in shape of "Deep", "Feedforward", and "Multilayer" and use it for creating the learning ability of the machine. The "deep" in "deep learning" refers to the number of layers but was not state clearly that how many layers should be considered to be "Deep". Deep learning usually forms in architectures such as deep neural networks, deep belief networks and recurrent neural networks (See Figure 10).

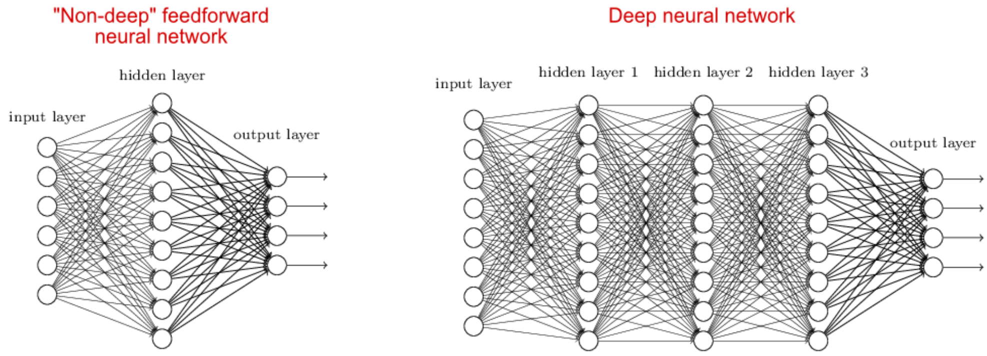


Figure 10. Deep neural network.

Image downloaded from http://neuralnetworksanddeeplearning.com/ in March 2019.

Several models of a Deep Neural Network have been developed for machine learning tasks. One of the challenging applications is the image object detection and recognition. Researchers have designed different structures of Deep Learning to work with the image but the most successful architecture is a "Convolutional Neural Networks" or "CNNs". This structure of a neural network is designed specifically for the image type data and used the advantage of a multi-layer network architecture to analyze the information out of the image. CNNs is the most often used algorithm for image recognition and has developed in a different model by many researchers today.

2.6 Convolutional Neural Network

Convolutional neural networks (CNNs, or ConvNet) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery. The basic structure of CNNs consists of several kinds of layer such as convolutional layer, pooling layer, fully connected layer where each layer has the different calculation in the network (See Figure 11).

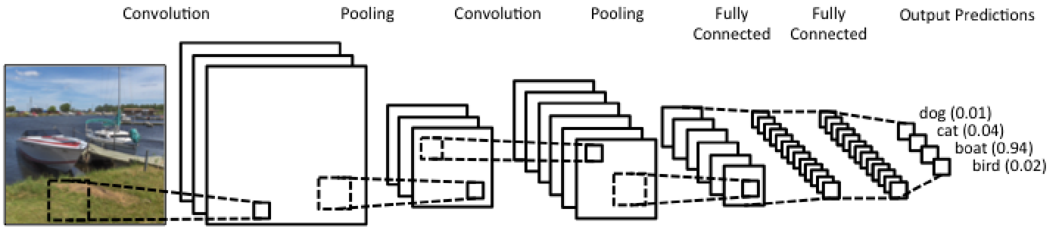


Figure 11. Structure of Convolutional neural network.

Image downloaded from https://towardsdatascience.com/ in March 2019.

Many researchers today use a convolutional neural network or CNNs largely in an area of image recognition and object detection because of the structure of convolutional layer that design specific for the input data format in 2d or 3d matrix structure like digital image format.

2.7 Deep Learning Model for Image Classification

2.7.1 AlexNet

AlexNet is the name of a convolutional neural networks created by Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever in 2010 (Krizhevsky et al., 2017). AlexNet was one of the pioneer approaches in terms of applying a CNNs into an area of image recognition. The model of Alexnet consisted of 650,000 neurons and 60 million parameters which had five convolutional layers and 1000-way softmax (See Figure 12). Alexnet was the winner on ImageNet Large-Scale Visual Recognition Challenge in 2010 (ILSVRC10) which was the competition on image recognition.

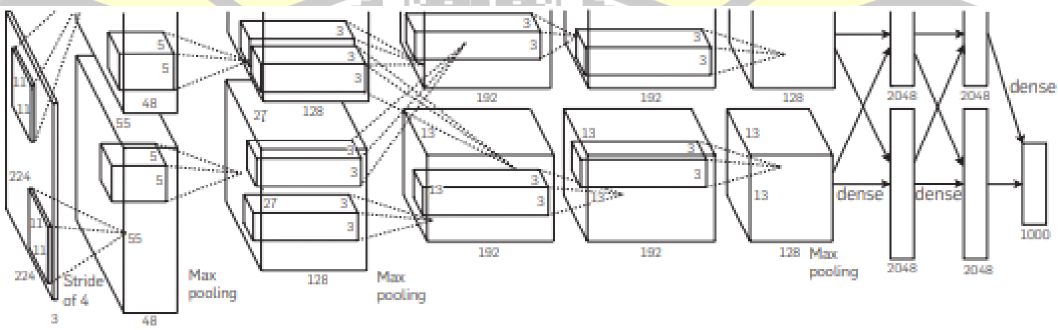


Figure 12. AlexNet architecture.

Image from Krizhevsky et al. (2017)

2.7.2 VGGNet

The VGG network architecture was introduced by Simonyan and Zisserman from Visual Geometry Group at University of Oxford in 2014 (Simonyan & Zisserman, 2014). This model is consisted of 3x3 convolutional layers, max pooling, two fully-connected layers, and softmax classifier (See Fig. 13). Simonyan and Zisserman had showed that the performance of VGG network was outperform the other models who was the winner of ILSVRC from year 2012 and 2013.

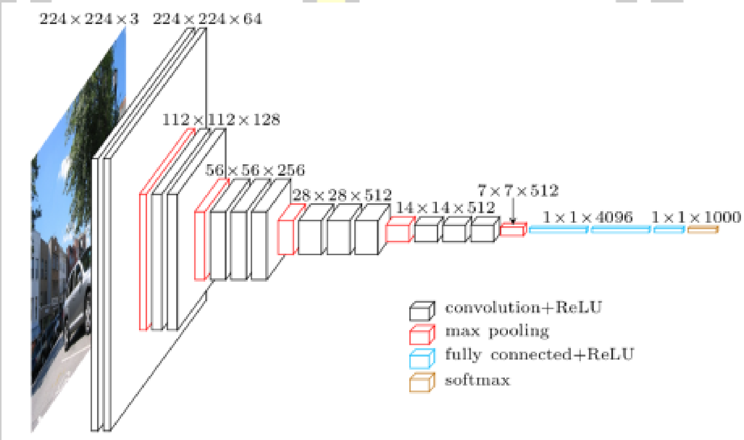


Figure 13. VGGNet.

Image from Simonyan et al. (2014)

The VGG model usually came in two different structures, VGG16 and VGG19. The number 16 and 19 means a number of weight layers in the network. VGG19 was perform better than VGG16 because of deeper layer but unfortunately increasing a lot of size for the whole network.

2.7.3 GoogLeNet

The GoogLeNet or Inception v1 is the name of a deep convolutional neural networks created by Christian Szegedy in 2014 (Szegedy et al., 2015). It consists of 22 layers neural network with the combination layers of convolution, max_pooling, softmax, and a new idea of inception module. The objective of inception layer is to find the optimal local construction in each layer and to repeat it spatially. Each "inception" module is the construction of the different sizes for each convolution node (1x1, 3x3, and 5x5) and 3x3 max pooling node (See Figure 14). GoogLeNet was the winner on ILSVRC14 which surpass the previous model such as AlexNet and VGGNet.

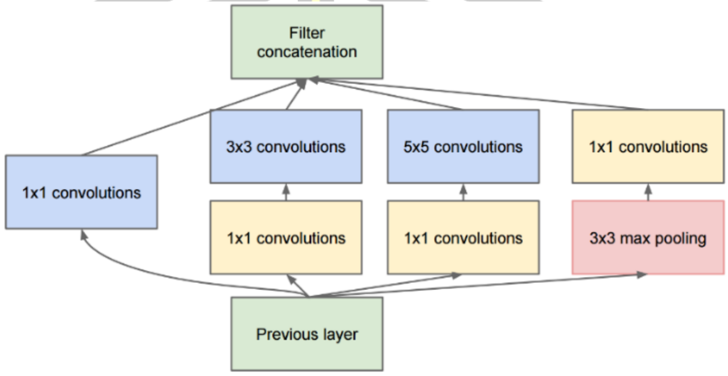


Figure 14. Inception module from GoogLeNet network (Inception v1). Image from Szegedy et al. (2015)

Follow the development on Inception v1, the improved version was released on the year later start from v2 to v4. Inception v2 (Ioffe & Szegedy, 2015) tried to improve the training performance of the networks by the premise of covariate shift. They did the batch normalization and replace a 5x5 convolution kernels with two 3x3 nodes. Inception v3 (Szegedy et al., 2016) factorized the convolution node in inception module and made a smaller size for convolution process. For example, they factorize one nxn kernel into nx1 and 1xn kernels (See Figure 15). And Inception v4 (Szegedy et al., 2017) is the redesign of Inception v3, which has more uniform simplified architecture and more inception modules. It accelerated the training module and also the performance of the network. The updated version of inception showed the better performance in each time it released. The latest v4 had the smallest error rate on ImageNet dataset compared to the previous three but require more parameters than version 3.

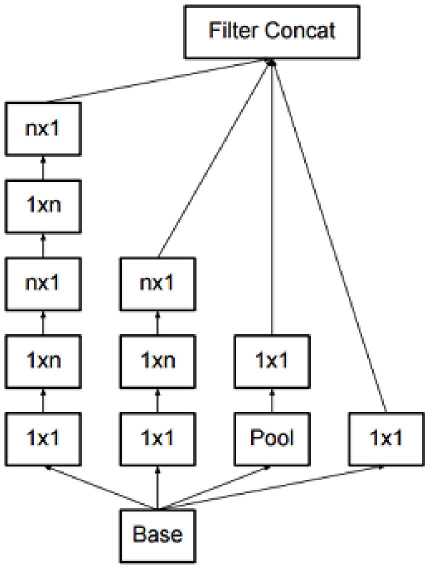


Figure 15. Factorization process in Inception v3.

Image from Szegedy et al. (2016)

2.7.4 MobileNets

MobileNets is another model of CNNs which has propose to reduce the size of the previous CNNs model to make it available to use in mobile platform (Howard et al., 2017). The idea is to replace the standard convolutional filters with two layers (Depthwise and Pointwise convolution) that build a smaller separable filter, which Depthwise reduce the size from length and width direction and Pointwise reduce the size of filter from depth direction (See Figure 16).

This network has achieved good performance compare to another model we mention above and come with the smallest size of the network. Make it a good choice for the researcher to deal with the problem of large-scale data.

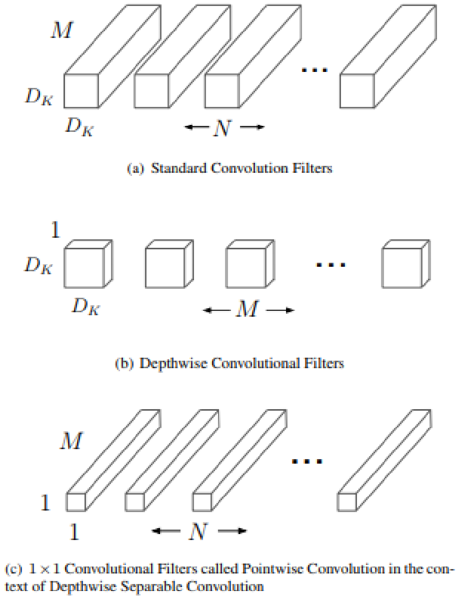


Figure 16. Depthwise and Pointwise separable convolution filter of MobileNets. Image from Howard et al. (2017)

2.8 Deep Learning Model for Image Detection

2.8.1 Region-Based Convolutional Neural Network (R-CNN)

The CNNs models we introduced in the previous section are mainly using for image recognition or classification problem. Those models can analyze the whole image and tell a human what the image object is it but cannot locate the location area of the object and more importantly cannot tell if there is more than one object in the image. The researcher took these issues for the next step, An Image Segmentation. The image segmentation (or Image Detection) is the process to find an area of the possible objects in the image which require to detect more than one object (if any) appears in the image. This step might include the image classification to define the type of the object in the same time or split into two different steps but mainly focus on capturing the area of the object before proceed to the next one.

Researchers in DL and CNNs have faced these issues as same as the people in another image processing groups. The CNNs was designed only for image classification but the obviously well performance of them forced many people to find the way to add an image segmentation into the CNNs. One of the pioneers in this task is the work of a UC Berkeley team, led by Ross

Girshick and Professor Jitendra Malik. It is called "Region-Based Convolutional Neural Network" or "R-CNN" (Girshick et al., 2015).

R-CNN was the combination of the classical image processing and the CNNs by using the method call "Selective Search" (Uijlings et al., 2013) to create a bunch of bounding boxes in the image with the high probability to be the same group of the object. These boxes (or some called Region of Interest (ROI)) will be analyzed by the search algorithm and choose the good region to be the input of CNNs for image recognition (See Figure 17).

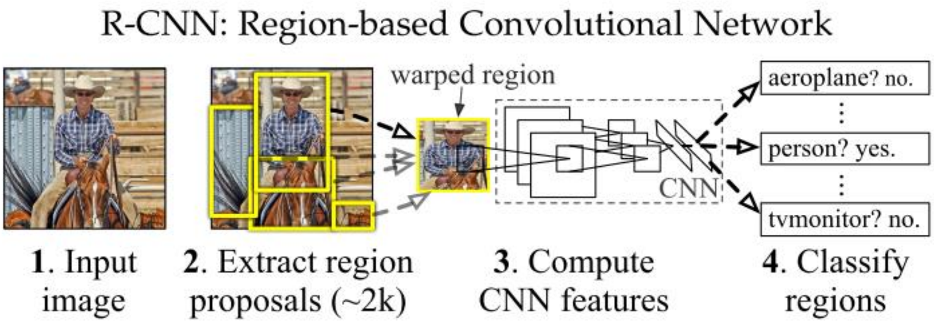


Figure 17. R-CNN create a set of a region bounding boxes before classification. Image from Girshick et al. (2015)

The combination features of Selective Search and CNNs showed an example of how to add the image segmentation into Deep Learning model for image detection. R-CNN was the early method that success in this issue and inspired many researchers to bring in more techniques of image processing to help CNNs overcome the problem of segmenting the objects in the image.

2.8.2 Faster Region-based Convolutional Network (Faster R-CNN)

Follow the success of R-CNN, another team at Microsoft Research in 2015, found a way to reduce the execution time of the region proposal step and named it "Faster R-CNN" (Ren et al., 2015). The main idea of Faster R-CNN is to replace "Selective Search" with the result of the early layers in CNN (that actually captured the similarity feature in the image and almost do the same thing with Selective Search). The Faster R-CNN using this first layer of CNN and extend it to be a Fully Convolution layer called "Region Proposal Network" by passing a sliding window over the CNNs feature map and at each window, and output the scores for how good of those boxes before sending to other layers (See Figure 18).

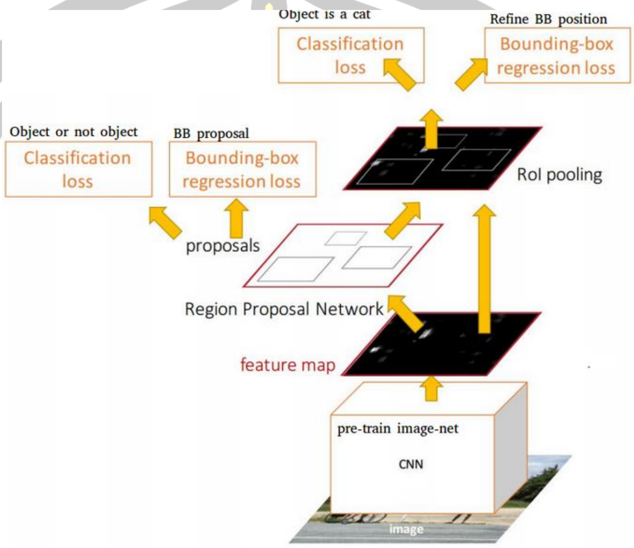


Figure 18. Faster Region-based Convolutional Network (Faster R-CNN). Image from Ren et al. (2015)

This replacement helps Faster R-CNN to surpass the old R-CNN in terms of speed and accuracy and pave the ways for others research group to add some region layer on top of CNN instead of using standard image processing method.

2.8.3 You Only Look Once (YOLO)

"You Only Look Once" or "YOLO" (Redmon et al., 2016) was another model follow the footstep of R-CNN. This model directly predicts bounding boxes and class probabilities within a single CNNs network by dividing an input image into an SxS grid. Each grid will predict a set of bounding boxes with a confidence score using a convolution layer (See Figure 19).

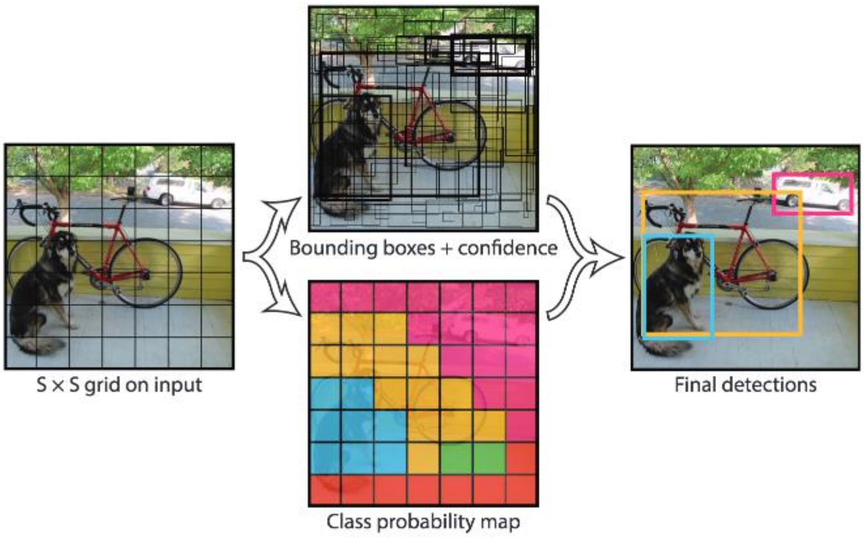


Figure 19. Example of SxS grid in YOLO.

Image from Redmon et al. (2016)

The YOLO CNNs network is inspired by the inception modules of the GoogLeNet model. This network has 24 convolutional layers and 2 fully-connected layers. The final layer outputs a tensor corresponding to the predictions for each cell of the grid. The Non-Maximum Suppression (NMS) (Neubeck & Van Gool, 2006) method is applied at the end of the network to reduce the number of boxes without objects (See Figure 20).

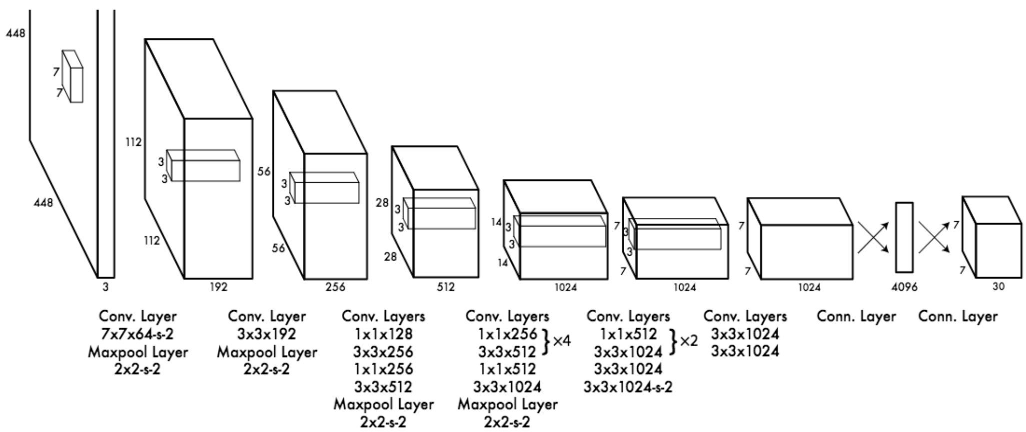


Figure 20. YOLO architecture.

Image from Redmon et al. (2016)

2.8.4 Single Shot Multibox Detector (SSD)

Single Shot Multibox Detector or SSD is similar to YOLO by creating one CNNs model that design to do the both tasks of image recognition (image segmentation and image classification) in the same time (Liu et al., 2016). The SSD model split an input image with different filter size (10x10, 5x5 and 3x3) to create a feature map using convolutional layer. These maps are used to predict the bounding boxes and processed by a specific layer with 3x3 filters called extra feature layers. Unlike YOLO, the SSD model uses these extra feature layers to increase the number of relevant bounding boxes (See Figure 21).

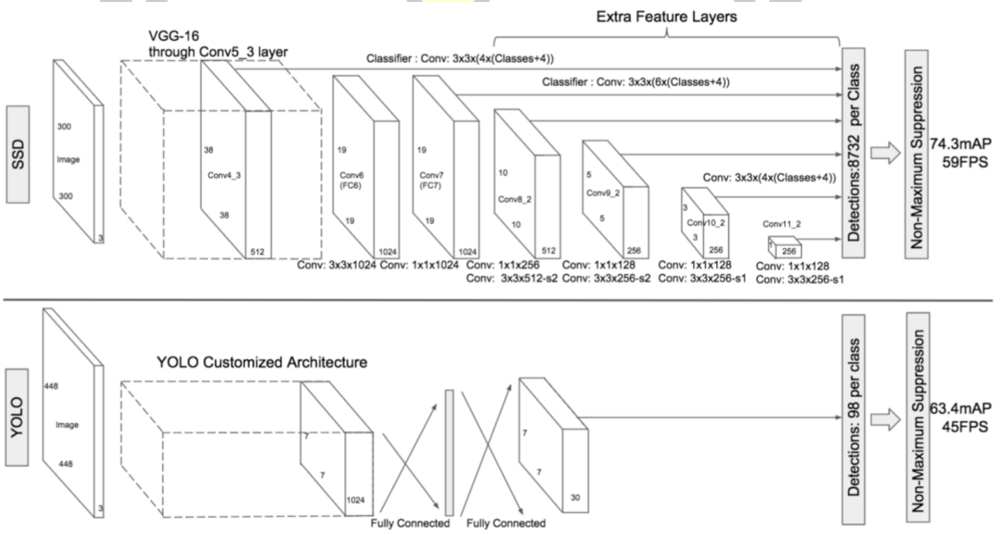


Figure 21. Comparison of the SSD and YOLO architecture.

Image from Liu et al. (2016)

SSD also uses the Non-Maximum Suppression to reduce the irrelevant bounding boxes like YOLO. But adding the Hard Negative Mining (HNM) to remove a lot of negative boxes that are still predicted.

2.9 Deep Learning Model for Image Restoration

2.9.1 Generative adversarial network (GAN)

A generative adversarial network (GAN) is a type of deep learning system based on two convolutional networks introduced by Ian Goodfellow and his team in 2014. The main concept comes with two neural networks contesting with each other in a task for generating new data with the same frequency as the training data. For example, GAN can generate a new image that looks closer to an original image dataset i.e., Human Face, Alphabet, Number, etc. (See Figure 22).

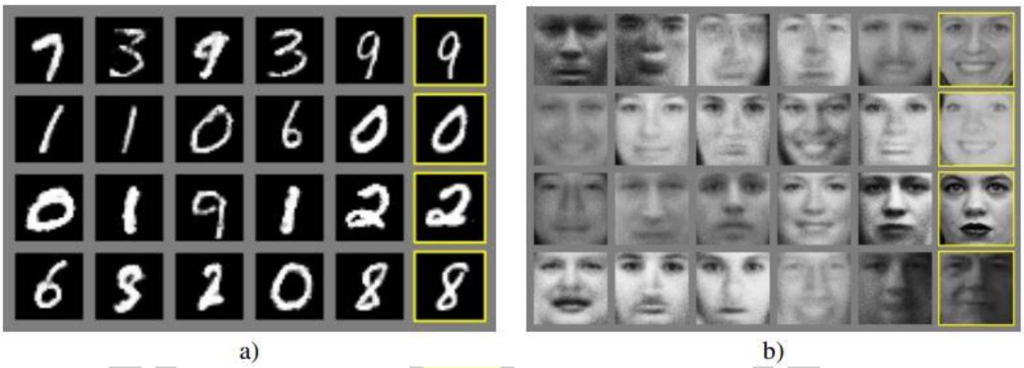


Figure 22. Example of generating images from GAN.

column 1-5 are the sample created from each distribution, column 6 is the training sample a) alphabet image b) human face image. Image from Goodfellow et al. (2014)

GAN structure consists of two CNNs. The first one called "a generative network" which tries to generate a new sample similar to the training data. And the second CNNs is a discriminative network that evaluates the performance of that sample. These two networks work together to help increase accuracy in their tasks. Finally, it generates a new output that seems like real data.

2.9.2 Generative adversarial network for Image Super-Resolution (SRGAN)

The application of GAN can also be used to create an upscaling image from the original photo. One example model is the Generative adversarial network for Image Super-Resolution or SRGAN (Ledig et al., 2017). Using the concept of two networks as same as the original GAN but adding a deep residual network inside standard CNNs (See Figure 23). SRGAN achieved the success of creating a 4 times factor upscaling image.

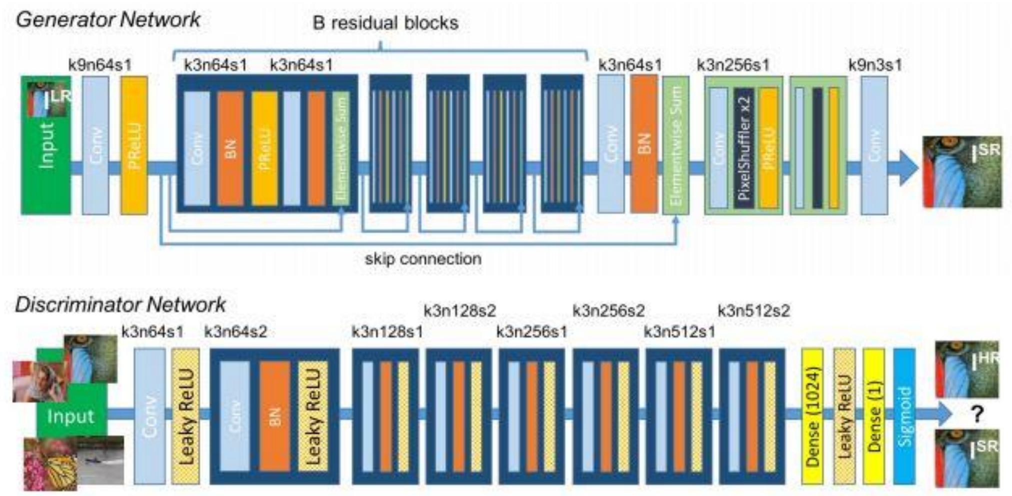


Figure 23. Structure of SRGAN.

Image from Ledig et al. (2017)

2.9.3 DeblurGAN

Another example of a GAN network that can improve image quality is DeblurGAN. In 2018 Orest Kupyn and his team used this network to recover a blurred image into a sharper output using two strided convolution blocks inside of GAN structure (See Figure 24). The result shows that this model can help increase the accuracy of the image recognition process using YOLO algorithm (Kupyn et al., 2018).

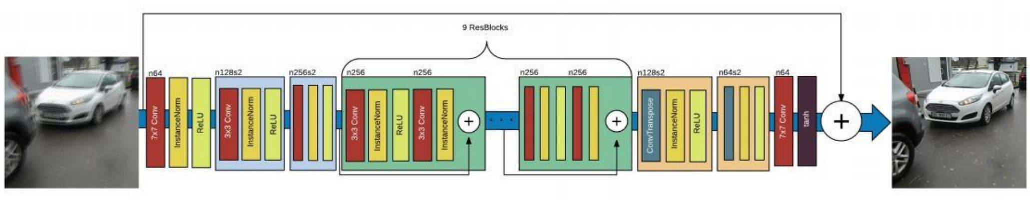


Figure 24. DeblurGAN.

Image from Kupyn et al. (2018)

2.10 Relevant Research

Zhou and Cheung (2016) researched the AlexNet model for vehicle type classification using the cropped vehicle image from the road image. They took feature vector from layers 6 and 7 of the pre-trained AlexNet model and used Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) for dimensional reduction. The result of classifying three types of vehicles (Sedan, Van, and Taxi) is 97%. But with a small size of image dataset (983 images, they could not fully retrain the network but require SVM for classification method instead (Zhou & Cheung, 2016).

Shaoyong Yu et al. (2017) developed a vehicle detection and classification using Faster R-CNN as the main algorithm for vehicle detection then used five layers CNN with a joint bayesian network to classify the model of the vehicle. Using 100,000 images of the vehicle acquired from the internet which consists of 73 vehicle brands. The result showed that the vehicle detection accuracy is only 85% and 89% on vehicle classification. The researcher found that the system still has a problem to distinguish the similar vehicle model and sometimes detect a non-vehicle object that might require to add more features in the classification model (Yu et al., 2017).

Zhanyu Ma et al. (2019) modified the standard max-pooling layer in Convolutional Neural Network to a new channel max-pooling (CMP) layer and experimented on a vehicle classification problem. This new layer tried to reduce unwanted features from the pool and help improve network performance. Its result showed the evolved accuracy of several CNNs models (VGG16, DenseNet161, and ResNet152) on standard Cars-196 and CompCars datasets (Ma et al., 2019).

Chen (2019) researched the Taiwanese license plate recognition using sliding-window darknet-YOLO architecture. The experiment ran on 2049 images of Taiwan car license plate which came from different locations, times, and weather conditions. His system achieves 98.22% accuracy on license plate detection but only 78% accuracy for character recognition. The researcher suggested that noise-reduction should be included in the process to improve the performance of the further experiment (Chen, 2019).

CHAPTER III

FAST AND ACCURATE DEEP LEARNING ARCHITECTURE ON VEHICLE TYPE RECOGNITION

Vehicle Type Recognition is a significant problem that happens when people require to search for vehicle data from a video surveillance system at a time that a license plate does not appear in the image. This chapter proposes to solve this problem with a Deep Learning technique called Convolutional Neural Network (CNN), which is one of the latest advanced machine learning techniques. In the chapter, researchers collected two datasets of Vehicle Type Image (VTID I & II) for 1,310 and 4,356 images. Then performed the first experiment with 5 CNN architectures (MobileNets, VGG16, VGG19, Inception V3, and Inception V4) and the second experiment with another 5 CNNs (MobileNetV2, ResNet50, Inception ResNet V2, Darknet-19, and Darknet-53) plus several data augmentation methods. The results show that MobileNets significantly outperformed other CNN architectures, which the highest accuracy rate at 95.46% when combine with the brightness augmented method and also the fastest model compares to other CNN networks.

3.1 Introduction

The increasing traffic problem around the world has forced several countries to build a system for assisting officers in controlling and securing their transportation problem. One of those countries is Thailand. Every year many cars are increasing the complexity of transportation control for the Thai government. Many researchers have tried to solve this problem for several years by adding a street video surveillance system to help police officers capture the illegal use of vehicles without them standing on the road as happened in the past.

Figure 25. Vehicle image without license plate detail.

The key problem of this system is how to detect a vehicle position in the image and how to extract the vehicle detail out from its license plate. But in many situations, the license plate does not appear clearly in the scene. That forces the system to identify other details of the vehicle instead. For example, vehicle shape, vehicle model, or vehicle type (car, truck, or bus), etc (see Figure 25). If the system can recognize these kinds of data, it helps to reduce the recognition processes on that vehicle. Many researchers brought these issues into the fields of image processing and machine learning to find a solution. Many state-of-the-art techniques have been applied to solve this problem, for example, Harris Corner Detector (Li et al., 2009), Histogram of Oriented Gradients (Zhang, 2012), or K-nearest neighbor (Clady et al., 2008). But these techniques require a long time for the training process and have not enough accuracy.

Fortunately, the development of another technique called "The Convolutional Neural Network" or "CNN", which bases on the deep neural network structure, has been proved to solve this problem. With the works of Dong (Dong et al., 2015), Huttunen (Huttunen et al., 2016), and Bautista (Bautista et al., 2016), they found CNN achieved better performance on vehicle type recognition problems compared to previous techniques. But still had a limit when the vehicle image came from a different viewing angle or direction. These problems need to be fixed with the improved CNN or another additional technique.

The first modern model of CNN is AlexNet (Krizhevsky, 2017), which proved that the idea of CNN would be the better method to solve several image recognition problems including the vehicle image. After AlexNet, other CNN models have been developed and achieved more signs of progress. For example, VGGNet (Simonyan, 2014), GoogLeNet (Szegedy, 2015), and

MobileNets (Howard, 2017). Each architecture enhanced the performance of CNN in different ways. For example, the recognition accuracy, the network structure, or the model parameters. Several problems in image recognition have been solving with this development (Boonsirisumpun & Puarungroj, 2018).

Even CNN has been a success in several areas of image recognition. They still face a significant accurate problem if the number of the training dataset is so small or the quality of a raw image is not satisfying. Then the addition of the data augmentation techniques requires creating more variance in the dataset and helps to improve the performance. From all problems have mentioned above, this research proposed to apply the CNN algorithms with the addition of several data augmentation techniques to the vehicle type recognition problem. To find the best combination method of these techniques on our low-quality vehicle image dataset.

Contributions: This chapter provides two new vehicle type image datasets, collecting from Loei Rajabhat University's video surveillance system. The first dataset consists of 1,310 images, name VTID. And the second has a total of 4,356 images (VTID2). Then apply ten convolutional neural network architectures, which are selected by the fast and accurate properties such as MobileNets (V1 and V2), VGGNet (16 and 19), GoogLeNet (Inception V3, V4, and Inception ResNet V2), ResNet50, and Darknet (Darknet-19 and 53) to challenge the dataset. Finally, the experiments of combination CNN and the data augmentation techniques have been performed.

Chapter Outline: The rest of the chapter is organized as follows: Section 3.2 describes the detail of several deep learning methods used in these experiments. Section 3.3 describes the data augmentation techniques. Section 3.4 introduces our image datasets. Section 3.5 describes the experiment details. The discussion part in Section 3.6, and finally, the conclusion is in Section 7.

3.2 Deep Learning

Deep Learning is a successful machine learning technique based on a deep layer neural network. This method has gained popularity in many areas in the field of artificial intelligence, machine learning, and image processing. The new architecture of deep Learning has developed, called a convolutional neural network or CNN, which is more suitable for digital large scale image data. One of the successful CNN pioneers is AlexNet, an eight layers deep convolutional network that first won the ImageNet Large-Scale Visual Recognition Challenge in 2010. After the success of AlexNet, many researchers applied the idea of CNN and developed their model to challenge the predecessor.

Convolutional neural networks (CNNs) are one of the most advanced algorithms for solving image recognition and detection. This pioneering idea dates back to the 1980s when it demonstrated how to apply CNN to analyze handwritten digit image (LeCun et al., 1989). The core structure of CNN is the Convolutional layer, a small filter that creates the feature map from the input image and then combines several features into interest objects (LeCun & Bengio, 1995). One famous example of the CNN model is the work of Alex Krizhevsky and his team during 2010-2012. He developed the full CNN model called AlexNet in order to compete in the ImageNet competition. This was followed by VGGNet from Simonyan and Zisserman in 2014, then Szegedy and a Google team introduced the GoogLeNet or 'Inception' between 2015 and 2017 (Szegedy et al., 2017; Szegedy et al., 2015; Szegedy et al., 2016). And in 2017, A.G. Howard created a small but efficient network called MobileNets.

3.2.1 MobileNets V1

MobileNets is a CNN model intended to reduce the size of the standard CNN model for making it suitable to use in a mobile device. The idea was to replace the general convolutional filters with two separate steps (Depthwise and Pointwise separable convolution) that build a lot smaller convolution filter. While Depthwise reduces the size from length and width dimension, Pointwise decreases the size of the filter from depth direction (Yoo et al., 2018). This network has achieved better performance compared to other large CNN models and had a smaller network size. These factors make it a suitable option for the researcher in dealing with the problem of large-scale training data.

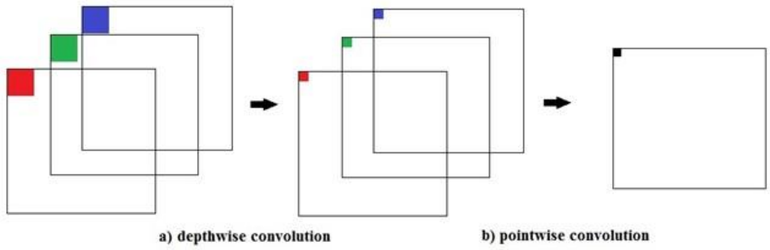


Figure 26. Illustration of the (a) Depthwise and (b) Pointwise separable convolution.

Depthwise and Pointwise separable convolutions are ideas for reducing the computation time in the CNN structure (Sifre & Mallat, 2014). This idea is of splitting the feature maps from one 3D convolutional layer into two 2D layers, first reduces the depth dimension, and then the height and the width (see Figure 26). This technique helps decrease the runtime and numbers of parameters, which makes it a small and fast CNN network.

Moreover, the designer of MobileNets has applied two more hyperparameters that can make the model even smaller; width and resolution multipliers. Width multiplier is a parameter to reduce the number of separable filters that split from the full convolution. For example, if choosing convolution size N x N x M. The standard MobileNets will separate them into M filters with N x N x 1 size. Width parameter (α) will replace it with αM filters instead.

The resolution multiplier parameter is the factor to directly reduce the size of the input image since the beginning of the first layer. By reducing the size of the start image, it helps to decrease the total computation filters along the way to the final layer.

The creator of MobileNets designed it to normally use four values for each multiplier. For the width multiplier, researcher can choose four values (1, 0.75, 0.5, and 0.25). And another four for the resolution multiplier (224 x 224, 192 x 192, 160 x 160, and 128 x 128).

3.2.2 MobileNets V2

The descendant of MobileNets, MobileNet V2, was introduced in 2018 (Sandler et al., 2018). The new version had two improvement concepts, Linear Bottleneck and Inverted Residual Block. The Linear Bottleneck layer was used to capture a linear transformation after the ReLU layer if some non-zero values are remaining (Figure 27). Inverted Residual Block is the idea to improve the standard residual block but has a thinner layer (bottleneck) instead of the thick one (see Figure 28). MobileNet V2 is a bit smaller than the first generation. The number of parameters reduce from 4.2 M to 3.4 M but still has an equivalent performance.

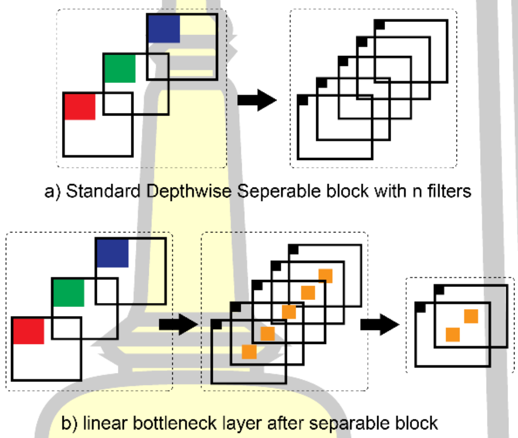


Figure 27. The structure of new separable block with linear bottle neck.
a) Standard separable block b) New separable block with linear bottleneck.

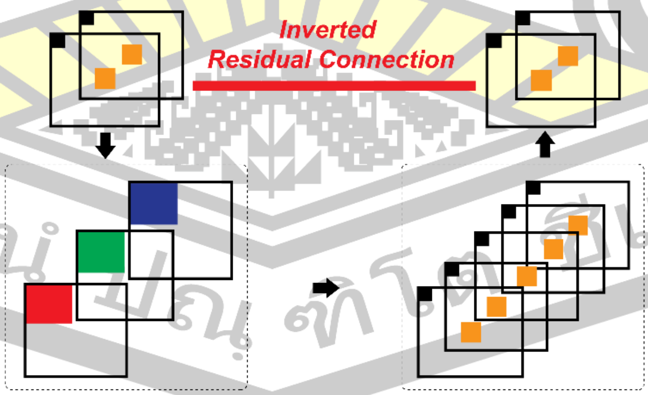


Figure 28. Inverted Residual Block connect to previous bottleneck layer instead of a convolution layers.

3.2.3 VGGNet

In 2014, Simonyan and Zisserman proposed VGG network architecture. This model used only 3×3 convolutional layers on top of the standard CNN network (max pooling, fully-connected, and softmax layer). Its results showed that the performance of the VGG network surpassed the other models at that time. The success of VGG architecture came in two sizes, VGG16 & 19. The number 16 and 19 refers to the numbers of weight layers in the network. VGG19 is better than VGG16 because of a deeper layer but unfortunately increased the size of the model.

3.2.4 GoogLeNet

GoogLeNet or Inception V1 is a CNN model created by Christian Szegedy and his team in 2014. In the first version, it had 22 convolution layers and an extra layer called the inception module. Each inception module is a construction of the different sizes for each convolution node (1×1, 3×3, and 5×5) and 3×3 max-pooling node. Inception V1 was the winner at ILSVRC14 and surpassed the performance of VGGNets.

After the success of Inception V1, GoogLeNet has implemented several updated versions following the predecessor. Inception V2 improved the training performance of the networks by using the batch normalization technique. The next model was the Inception V3. This time they factorized the convolution node in the Inception module and made it smaller. The latest version is Inception V4. This network is a redesign of Inception V3 by adding more uniform architecture and more inception modules. The performance showed that Inception V4 had the smallest error rate on several datasets compared to the previous three networks but required more runtime than Inception V3.

3.2.5 ResNet 50

ResNet is the first CNN using the concept of residual connection that was introduced in 2015. Its concept is to create a shortcut connection using residual learning block (Figure 29). The result of these shortcuts was proposed to reduce training error for a deep layer CNN model and it successfully outperformed against the VGG network at that time (He et al., 2016).

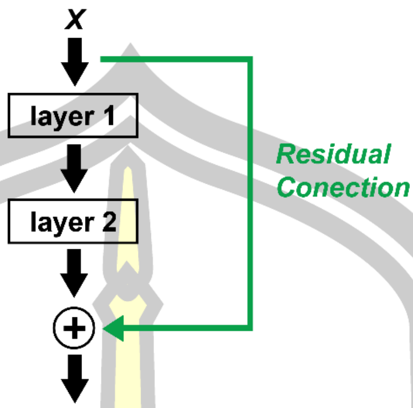


Figure 29. Residual learning block create a shortcut connection (green arrow) to reduce the path from input to output layer.

3.2.6 Inception ResNet V2

Inception ResNet V2 is the combination concept of inception network and residual connection. It was originally introduced along with inception V4 and Inception ResNet V1 with the proposal to use the residual connection to lessen the training process of Inception V3. While V4 is a larger version of Inception V3, Inception ResNet V1 is the combination of V3 and residual connection, and Inception ResNet V2 is the V4 plus residual block idea. From the experiment of Ioffe & Szegedy in 2015, Inception ResNet V2 had won the best performance compared to the other three in terms of accuracy. It had improved training speed compared to inception V4 (Ioffe & Szegedy, 2015).

3.2.7 Darknet-19 & Darknet-53

Another interesting CNN network is called "Darknet". The core structure of the object detection technique "YOLO" (Redmon et al., 2016). YOLO is one of the pioneers in using only one neural network to predict both object location and object type in the image. The architecture of YOLO can be separated into two parts, first the bounding box detection and second the image classification. Originally, YOLO V1 used almost the same structure as GoogLeNet (Creating inception module with convolution layer).

YOLO V2 was followed in 2017 with the new classification part called "Darknet-19" (Redmon & Farhadi, 2017). Darknet-19 architecture is mostly similar to VGG19 (19 convolution layer) but the special module that seems like an inception block inside the convolution layer.

Another model derived from the YOLO series is "Darknet-53". This is the same as Darknet-19, Darknet-53 is the main structure of YOLO V3 with a bigger size (53 convolution layers) has a residual connection like Inception Resnet. The YOLO V3 was reported to be slightly worse than a previous version but it returned a good result on the detection metric of .5 IOU (Redmon & Farhadi, 2018).

3.3 Data Augmentation

Image classification with deep learning was performed with a good result but required a large amount of training data to avoid overfitting (Shorten & Khoshgoftaar, 2019). In some situations, researchers can collect enough data to train the model. But in many situations, this is not possible because of a lack of resources. One method to solve this problem is data augmentation, a technique that attempts to create more data samples by adjusting or transforming the original image in various ways.

Data augmentation can be accomplished both with some basic transformations and by using other advanced methods to generate a larger training dataset (Mikołajczyk & Grochowski, 2018). Some popular basic augmentations are the affine transformations (Flipping, Rotating, Zooming, and Shifting). These techniques try to create a new image using basic image transformation, for example, move the pixel, reflect the image vertically or horizontally, and rescaling or cropping the image. Another way is performed using color modification. Researchers can adjust an image by changing its color system, converting the color image into a black-white image or vice versa, and enhancing the contrast or brightness of the image.

3.4 Dataset

3.4.1 Vehicle Type Image Dataset (VTID)

The main objective for the image dataset is to examine five vehicle types of Cars that are mostly used in Thailand (sedan, hatchback, pick-up, SUV, and van). The recording devices to collect the image are from a video surveillance system of Loei Rajabhat University in Loei province, Thailand. The collection process took place during the daytime for four weeks between July and December 2018. Two cameras were installed at the front gate of the university. However, with a small number of the Van images in the dataset compared to the other four vehicle types. In this case, the researcher decided to add other vehicle-type images such as a Motorcycle in the Van group and change the name to “Other vehicles” (Figure 30) to increase diversity. Finally, the first dataset called "Vehicle Type Image Dataset (VTID)" has a total of 1,310 sample images that can be separated into each vehicle type as follows; 400 Sedans, 478 Pick-ups, 129 SUVs, 181 Hatchbacks, and 122 Other vehicle images.

Each image has been collected using the 224x224 resolution. With a total number is more than 1,000 images, the overall images are exceedingly more than a million parameters. Make it more suitable for using CNN than other techniques to apply for this dataset.

Figure 30. Example of VTID1 dataset collected in four different views (front, back, left, and right): a) Sedan, b) Hatchback, c) Pick-up, d) SUV, e) Others vehicle.

3.4.2 Vehicle Type Image Dataset 2 (VTID2)

After VTID, researchers decided to extend the collection process to create another larger dataset to add diversity to the dataset for avoiding data overfitting. Finally, the new dataset, call "Vehicle Type Image Dataset 2 (VTID2)", consists of 4,356 image samples that can be separated into five vehicle type classes as follows; 1230 Sedans, 1240 Pick-ups, 680 SUVs, 606 Hatchbacks, and 600 Other vehicle images.

Moreover, the five data augmentations (Horizontal Flip, Random Shift, Rotation, Zoom, and Brightness) have been applied to the VTID2 dataset and creating five larger datasets in the experiment (VTID2-Flip, VTID2-Shift, VTID2-Rotation, VTID2-Zoom, and VTID2-Brightness) (Figure 31). These five datasets were generated using the Keras ImageDataGenerator library (Chollet, 2016) with parameter settings for Shift range (-50 to 50), Rotation degree (-45 to 45), Brightness (0.2 to 1.0), Zoom range (0.5 to 1.0), and used in the experiment first separately, then combining (VITD2-All) in the final test.

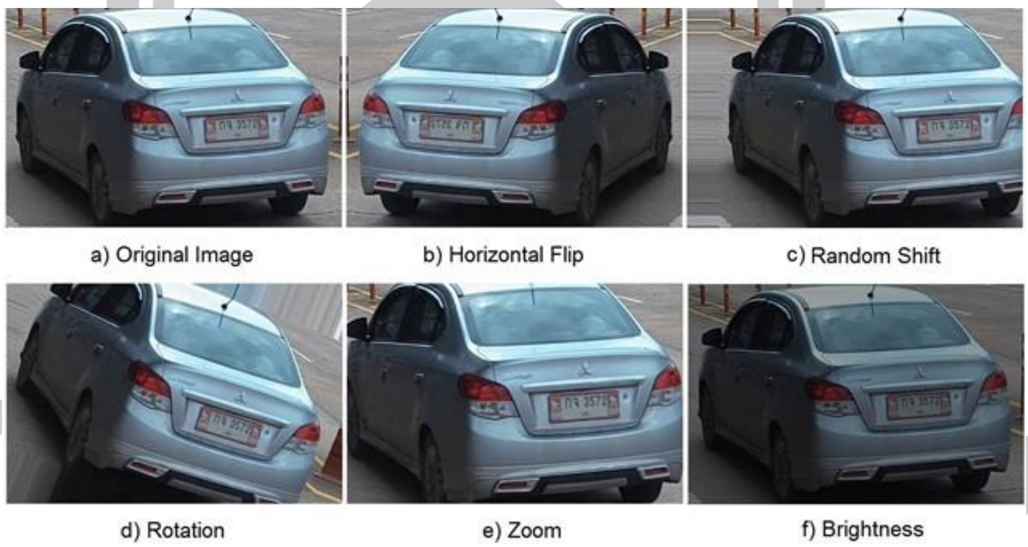


Figure 31. VTID2 with data augmentation (a) original image, (b) horizontal flip, (c) random shift, (d) rotation, (e) zoom, and (f) brightness

3.5 Experiments

3.5.1 Experimental settings and results

For the experiment, the model evaluation has been designed base on 10-fold cross-validation for comparing both the accuracy (Eq.1) and standard deviation on the first VTID dataset. The dataset has been separated into training, validation, and test sets, containing 917, 131, and 262 images, respectively. Then, five types of vehicle images; sedan, pick-up, SUV, hatchback, and other vehicles are divided randomly into each set.

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+TN+FP+FN}} \qquad (1)$$

where

$$TP = True\,positive;\; FP = False\,positive;$$

$$TN = True\,negative;\; FN = False\,negative$$

Then, several standard CNN architectures will be applied to perform image classification experiments on VTID. The overall experiments were run using Python 3.7 with TensorFlow and Keras library on one CPU (Intel Core i-7, 8th Gen, 4.0 GHz, Ram 8 GB). All training processes were trained from scratch for 20 epochs without using any pre-trained models to observed the real accuracy and training time of each model.

3.5.2 Experimental with the MobileNets

In the first part of the experiment, the test has been run only on MobileNets architecture but with a different value of its two hyperparameters, The Width and Resolution multipliers. The first results in Table 1 show the performance obtained from MobileNets with vary Width Multiplier on the VTID dataset. The result found that the number of width multiplier directly affects the accuracy performance and the size of the model. In this case, the size of the model decreases heavily related to the value of the width multiplier (From 16, 10, 5, and 2 MB). For this reason, the accuracy of vehicle type recognition is also steadily decreased from 93.40% to 84.33%.

Table 1 Performance of MobileNets on different width multipliers (Fixed Resolution = 224).

Width Multipliers	Accuracy (%)	Size (MB)
1.0	**93.40**	**16**
0.75	88.10	10
0.5	84.33	5
0.25	86.66	2

Table 2 shows the results based on another hyperparameter, the Resolution. The results were different from the previous one, but the best value is still the combination of 224x224 resolution and 1.0 width multiplier. However, the size of the resolution value of the input image does not directly affect the size of the model as same as that happened for the width parameter.

Table 2 Performance of MobileNets on different resolution multipliers (Fixed Width = 1.0).

Resolution	Accuracy (%)	Size (MB)
224 x 224	**93.40**	**16**
192 x 192	92.54	16
160 x 160	91.20	16
128 x 128	90.39	16

The extension of the Mobinet experiment has been performed with every combination of these two hyperparameters. The results are presented in Table 3 confirm that the 224x224 resolution and 1.0 width multiplier is the best combination for the MobileNets network for the VTID dataset.

Table 3 Accuracy of MobileNets on every combination of resolution and witdh multipliers.

Resolution	Width Multipliers			
	1.0	0.75	0.5	0.25
224 x 224	**93.40**	88.10	84.33	86.66
192 x 192	92.54	87.27	83.21	85.24
160 x 160	91.20	86.66	82.67	84.33
128 x 128	90.39	84.33	82.72	83.54

The analysis of the MobileNets performance has been shown in the form of the confusion matrix as seen in Table 4. The highest error type of vehicle was SUV with the lowest accuracy of only 70.54%. The main mistake of the SUV probably because it has more similarity to the other three vehicle types group The SUV has front and side shapes that are close to sedan and pick-up, and most of its shape (front, side, and rear shape) is quite similar to a hatchback.

Table 4 The average confusion matrix of the MobileNets architecture.

Actual Class	Prediction Accuracy (%)				
	Sedan	Pick-up	SUV	Hatchback	Other Vehicles
Sedan	**95.25**	1	0.75	3	0
Pick-up	0.21	**98.33**	1.46	0	0
SUV	6.98	11.63	**70.54**	10.85	0
Hatchback	9.39	1.66	1.11	**87.84**	0
Other Vehicles	0	0	0	0	**100**

The examples of SUV's error are shown in Figure 32. The first row is the example of SUV type that has been classified to be Pick-Up (Figure 32a), then Sedan (Figure 32b), and finally Hatchback (Figure 32c).

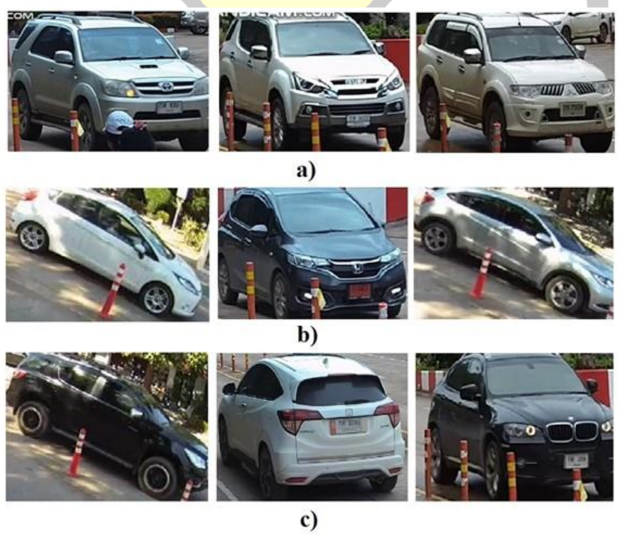


Figure 32. Examples of the wrong result of the SUV when predicting to be (a) Pick-up, (b) Sedan, and (c) Hatchback.

3.5.3 Comparison of the MobileNets and Other CNN Architectures

In the second experiment, The MobileNets will be challenged to the other four predecessors CNN (VGG16, VGG19, Inception V3, and Inception V4) on the VTID dataset. The best parameter values from the previous MobileNets experiments (resolution multiplier = 224 × 224 pixels and width multiplier = 1.0) were selected. The results are shown in Table 5. The results show that MobileNets architecture significantly outperforms all other four CNN methods.

The results in Table 5 show that MobileNets architecture significantly outperforms four large-scale CNN architectures (Inception V3 and V4 and Vgg16 and 19) in terms of accuracy, image resolution, and size of the model. To summarize the results, the accuracy of MobileNets is 93.40% followed by Inception V4, Inception V3, VGG19, and VGG16 with accuracies of 90.36, 88.81, 79.77, and 77.53%, respectively. The results also show that MobileNets overcame the performance of Inception V4 more than 3% and has the lowest uncertainty with a standard deviation rate of only 0.95.

Table 5 The average recognition accuracy and the standard deviation of the CNN architectures computed on our vehicle type image dataset.

Network Model	Image Resolution (Pixel)	Accuracy ± S.D.	Size (MB)
MobileNets	**224**	**93.40 ± 0.95**	**16**
Inception V4	299	90.36 ± 1.58	163
Inception V3	299	88.81 ± 1.80	83
VGG19	240	79.77 ± 2.04	548
VGG16	240	77.53 ± 2.22	527

In terms of model size, MobileNets was also the best model with the smallest size. The total size of the MobileNets network is only 16 MB which is ten times smaller than Inception V4 (163 MB). Another interesting issue is that the MobileNets outperforms Inception V4 in accuracy even with a smaller size. However, Inception V4 needs to increase size by doubling the size of Inception V3 (83 MB) to obtain higher accuracy than Inception V3.

For the VGG networks (VGG16 and VGG19), VGG19 shows better accuracy than VGG16 but needs a larger network. This is the same issue with the Inception networks as mentioned above. However, the size of both networks is large, 548 MB for VGG19 and 527 MB for VGG16, while the accuracy of VGGNet was the lowest amongst all five networks making it a poor choice in this problem.

An average runtime on five CNN networks (MobileNets, Inception V4, Inception V3, VGG19, and VGG16) shown in Table 6. The experiments found that MobileNets was the fastest model both in training time and test time with average runtime about 18.25 min on training and 7.29 min on the testing process. The second fastest is Inception V3 with training runtime 22.40 min and testing 10.21 min. Inception V4 is the third-place speed model with 27.53 min and 18.30 min on training and test. Moreover, VGGNets (16 and 19) still are the poorest networks with the slowest runtime compare to the other three networks around 30 min both in training and test processing.

Table 6 Evaluate the runtime performance of CNN networks on our dataset.

Network Model	Training Time (Min.)	Test Time (Min.)
MobileNets	**18.25**	**7.29**
Inception V4	27.53	18.30
Inception V3	22.40	10.21
VGG19	37.21	29.56
VGG16	35.38	27.22

From the result shown in Tables 5 and 6, MobileNets outperformed the other four networks in vehicle type recognition problem in every term (accuracy, size, and speed).

3.5.4 Experiments with Data Augmentation

In the third and final experiment, the performance of MobileNets and other CNN models will be challenged with two main proposes. First, the higher diversity of the vehicle type image dataset with larger data examples (VTID2) and additional data augmentation techniques. Second, the newer CNN models with concepts of Residual Block Connection.

For this section, the experiments have ran on seven image datasets that were created after adding the data augmentation techniques as mentioned in section 4.2 (VTID2, VTID2-Flip, VTID2-Shift, VTID2-Rotation, VTID2-Zoom, VTID2-Brightness, and VTID2-All) using 8 convolutional neural network architectures: MobileNets, Inception V3, Inception V4, ResNet50, Inception ResNet V2, Darknet-19, Darknet-53, and MobileNetV2 (excluding VGG16 and VGG19) using 10-fold cross-validation as in the previous experiment. A standard VTID2 has 4,356 images, while the other data augmentation dataset was double the size of the first one. Finally, the VTID2-All combined all pictures from every data augmentation method to make the total number of images of 26,136 (6 times of VTID2).

The results of data augmentation showed some surprises in the performance of the first MobileNet (See Table 7). In terms of total accuracy, MobileNets outperformed the other seven networks in every dataset. MobileNetV2 and Darknet-53 were the closest competitors for each other while MobileNetV2 won 5 of 7 against the opponent. Inception Resnet V2 overcame the performance of both Inception V3 & V4 and ResNet50 but still followed the lead of the MobileNets and Darknet series.

Table 7 The average recognition accuracy (%) using data augmentation.

Network Model / Dataset	VTID2	VTID2 Flip	VTID2 Shift	VTID2 Rotation	VTID2 Zoom	VTID2 Brightness	VTID2 All
MobileNets	**94.38**	**95.38**	**94.98**	**93.93**	**94.19**	**95.46**	**93.98**
Inception V4	92.17	90.77	90.02	87.28	88.07	91.14	86.40
Inception V3	91.37	91.45	89.49	88.06	86.83	91.01	87.95
ResNet50	91.24	91.33	89.57	87.85	87.02	90.95	87.65
Inception ResNet V2	92.95	93.71	94.27	89.86	90.01	94.65	89.93
Darknet-19	91.58	91.98	91.32	88.65	87.04	92.07	89.13
Darknet-53	93.99	94.54	94.02	91.87	92.01	94.98	92.87
MobileNetV2	93.56	94.32	94.65	92.85	92.89	95.22	93.60

For the effect of data augmentation, the best augmentation-technique was the brightness method. It had the highest score in 5 of 7 models excluding only Inception V3 & V4. MobileNets had the highest accuracy (95.46%) with data brightness. The worst method in our experiment was Image Rotation. It had the lowest score but also, and surprisingly the same ratio (5 of 7).

Unfortunately, combining every data augmentation (VTID2-All) does not show significant improvement in the VTID2 dataset. The accuracy of VTID2-All was around the lowest score in every CNN architecture compared to using only one augmentation.

For the training runtime (see Table 8), the MobileNets family is still the fastest model. At this time, MobileNetV2 is the most active network because of its smaller size compared to its parent. Inception ResNet V2 is a little slower than Inception V4 but with higher accuracy. ResNet50 was slightly slower than Inception V3 but has a close performance on accuracy. The speed of Darknet-19 is near the same as MobileNets but its accuracy is different.

Table 8 Evaluate the training time performance of CNN networks after data augmentation.

Network Model	VTID2 (min.)	VTID2-ALL (min.)
MobileNets	50.72	259.24
Inception V4	81.54	453.63
Inception V3	65.87	359.48
ResNet50	74.21	402.56
Inception ResNet V2	91.14	502.25
Darknet-19	53.22	284.61
Darknet-53	102.32	547.93
MobileNetV2	**42.18**	**215.26**

Finally, the new confusion matrix for the VTID2 dataset and Brightness augmentation (Table 9) shows the impressive improvement of each vehicle classification rate. Especially, on SUV and Hatchback. The accuracy of SUV rises from 70.54% to 88.23% and for Hatchback is increase from 87.84% to 92.07%. It confirms that increasing the size of the image dataset and using data augmentations techniques can help to improve the overall performance of image classification.

Table 9 The new confusion matrix of VTID2 with MobileNets architecture and Brightness augmentation.

Actual Class	Prediction Accuracy (%)				
	Sedan	Pick-up	SUV	Hatchback	Others Vehicle
Sedan	**96.75**	0.8	0.5	1.95	0
Pick-up	0.18	**98.39**	1.43	0	0
SUV	2.95	4.85	**88.23**	3.97	0
Hatchback	5.81	1.47	0.65	**92.07**	0
Others Vehicle	0	0	0	0	**100**

3.6 Discussion

The results from the previous section show the outstanding performance of MobileNets in these vehicle type recognition experiments. This model has received the highest accuracy in both datasets (VTID and VTID2) and also requires the least training time to create a model.

However, to compare the real accuracy and runtime of each CNN model. The researchers decided to run the experiment by training each model from the scratch without bias from the pre-trained parameters and train only 20 epochs to compare its speed. These might be the issues to study further if the pre-trained values or longer training process will affect the result.

3.7 Conclusion

In this chapter, the experiments on applying deep learning models for solving the issues of vehicle type image recognition have been performed. Ten convolutional neural networks (CNNs) had been chosen to use in this chapter (MobileNets, VGG16&19, Inception V3, Inception V4, ResNet50, Inception ResNet V2, Darknet-19, Darknet-53, and MobileNetV2) to compare the performance of each architecture.

The results found that MobileNets is the best method to deal with this vehicle type recognition problem all in terms of speed, accuracy, and size. The accuracy of MobileNets is the best, the model size is the smallest, and its runtime is the fastest compared to the other architectures. Considering that MobileNets is the best, fastest, and smallest size made it suitable to be used in every computing platform, including mobile devices with having a slower speed and smaller memory.

Additionally, the SUV vehicle type is the worst model to classify because of the similarity in its shape compared to other vehicle types, including Pick-up and Hatchback. Incidentally, this might be affected by the unbalanced amount of each image class in the training data. It requires the further work to create a new unbiased dataset with a larger size and the same number of images in every vehicle type.

Another interesting result is MobileNets' parameter-tuning. It found that both width multiplier and resolution multiplier did not improve the chances to defeat this problem. These two parameters can help to reduce the MobileNets size, but do not enhance accuracy.

Moreover, with data augmentation, the result found that MobileNets still wins against other CNNs in every single augmentation technique. The best augmentation method found in our research was Image Brightness. The combination of our second dataset (VTID2), brightness augmentation, and MobileNets V1 helped reach the highest accuracy (95.46%).

For future work, it is interesting to add some improvement inside the architecture of MobileNets (or other models) to increase the accuracy rate and reduce its runtime when dealing with another larger dataset.

CHAPTER VI

ENSEMBLE MULTIPLE CNNs METHODS WITH PARTIAL TRAINING SET FOR VEHICLE IMAGE CLASSIFICATION

Convolutional neural networks (CNNs) are now the state-of-the-art method for several types of image recognition. One challenging problem is vehicle image classification. However, applying only a single CNNs model is limited due to the weakness of each model. This problem can be solved using the ensemble method. Using the power of multiple CNNs together helps increase the final output accuracy but is very time-consuming. This chapter introduced the new "Ensemble Multiple CNNs methods with Partial Training Set" method. This method combines the advantages of the ensemble technique to increase the recognition accuracy and uses the idea of a "Partial Training Set" to decrease the time of the training process. Its performance help to decrease by more than 60% of the time-consuming process but it is still able to maintain a high accuracy score as 96.01% compared to the full ensemble technique. These properties make it a good choice to compete with other single CNNs models.

4.1 Introduction

Vehicle image classification is an important issue in the world of computer vision. A benefit from understanding information about each vehicle is that it is possible to solve problems in intelligent transport and security systems. For example, controlling the traffic, detecting a specific car, tracking the movement of vehicles, or eventually guiding a self-driving car on the road. This issue can be separated into many problems due to the complex features of vehicle images such as vehicle type, vehicle shape, vehicle color, vehicle model, vehicle make (logo), and vehicle size.

Many algorithms have been chosen to solve these problems. One modern state-of-the-art method is convolution neural networks (CNNs), the complex machine learning model based on a deep neural network. Several successful CNNs models have been introduced since the 2010s. For example, AlexNet (Krizhevsky et al., 2012), VGGNet (Simonyan and Zisserman, 2014), GoogLeNet Inception (Szegedy et al., 2015), ResNet (He et al., 2016), and MobileNets (Howard et al., 2017). Each model had its advantages and effects on many kinds of image recognition problems.

However, even though the performance of CNNs is quite acceptable for image recognition tasks but some problems remain and are described below. The first problem is choosing the best model for the dataset. Selecting only a single CNNs model that runs strongly on each dataset is uncertain. Some datasets probably have more effect on the complex models while others operate properly on simple models. These are challenging. Finding a good match between model and data requires a lot of experiments to be performed. This problem inspired the idea of using multiple models at the same time to help fix the weak point of each model in the prediction. A uniquely effective idea is the ensemble method. This method can use multiple learning models to run the prediction separately at the same time and then combine the result from a different model or different data to help predict a more accurate and more solid results (Re & Valentini, 2012).

The second problem is the cost of the time taken. CNNs usually require a massive size of training dataset and a very long time to train the model. These problems are tough for the simple CNNs model and even worse for the ensemble method because their use of multiple CNNs together on the full size of the training dataset requires exponential amounts of time in the training process. This chapter proposed a solution for this issue by using only some parts of the training set, which were called the partial training set, for each CNNs instead of the complete set. This chapter intended to show the performance of the ensemble method with multiple CNNs models for vehicle image classification.

4.2 Materials and Methods

4.2.1 CNNs

CNNs are the modern algorithm for image recognition that found success, starting in the 2010s. For example, AlexNet, VGGNet, GoogLeNet or Inception, ResNet, and MobileNets. Each model had a different structure, size, and performance on the image classification problem. This research chose five recently state-of-the-art models to perform the experiment with the ensemble method.

1) MobileNets

MobileNets is a tiny CNN model (4.2 M parameters) using the concept of depthwise and pointwise separable convolution to reduce the model size to be suitable for a mobile platform (Howard et al., 2017). The second-generation (MobileNets V2) followed in 2018 (Sandler et al., 2018) and was somewhat smaller than the previous one (3.4 M parameters). Both models were good in terms of accuracy and speed.

2) GoogLeNet

GoogLeNet or Inception was created by Christian Szegedy in 2014. The recently stable version was V3 and V4 (Szegedy et al., 2016 and 2017). This model could be considered as a medium-size CNNs method with 24 M parameters, which was a lot smaller, compared to AlexNet, but with higher performance. GoogLeNet was one of the standard models used in image recognition tasks (Szegedy et al., 2015).

3) ResNet50

ResNet was introduced by Kaiming He in 2016, choosing the residual learning block as its core structure (He et al., 2016). This chapter chose this model for the experiment as another medium-size CNNs example.

These three methods found success in several vehicle image recognition approaches. For example, Špaňhel applied MobileNets and Resnet50 in vehicle type and color recognition, which improved the accuracy of low-power devices (Špaňhel et al., 2018). Puarungroj studied the performance of Inception-V3 and performed experiments on vehicle license plate images (Puarungroj & Boonsirisumpun, 2018). Thomas used the combination of Inception and Resnet model for the recognition on moving vehicle (Thomas et al., 2020), while Goh implemented the transfer learning MobileNets and achieved higher accuracy and low latency on a real-time vehicle dataset from a video surveillance system (Goh, 2021).

4.2.2 Ensemble Multiple CNNs Methods

Ensemble methods are learning algorithms that combine a set of model classifiers and then take the vote or weight summary of their predictions to make a final answer (Polikar, 2012). The original technique used was Bayesian averaging (Dietterich, 2000). Ensemble methods can be used to connect different kinds of model predictors, such as binary tree, support vector machine, and neural network. Each model predicted its own result and then used several ways to combine their prediction. The examples of the combination technique are techniques such as majority voting (Raza, 2019), weight average (Dogan & Birant, 2019) and unweighted average (Sewell, 2011). This research proposed using the two simplest ensemble techniques, the unweighted average method and the majority vote method, with several CNNs classifiers.

1) Unweighted average method

The unweighted average method is the ensemble method that computes the final prediction of multiple CNNs models by summarizing the probabilities of all models and then dividing that by the number of the models (averaged probability). This method gave the final prediction from the highest probability answer as a result, which can be defined by equation (1):

$$\hat{y}_i = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{1}$$

where y_i is the output probabilities of each CNNs model and n is the number of the models.

2) Majority vote method

The majority vote method is the simple ensemble method that computes the final prediction of multiple CNNs models by directly counting the result of each model using the argmax function. Then the maximum vote was decided which can be defined by equation (2):

$$\hat{y}_i = \frac{1}{n}\sum_{i=1}^{n} argmax(y_i) \tag{2}$$

where y_i is the output probabilities of each CNNs model and n is the number of the models.

4.2.3 Ensemble Multiple CNNs Methods

Ensemble multiple CNNs methods were effective ways to summarize the prediction from many CNNs classifiers. They gave higher accuracy of prediction but unfortunately, were compromised by having much longer training times. Because of using more than one classifier, each CNNs needed to be trained separately with a full-size training dataset.

4.2.4 Dataset

These experiments used two types of vehicle image datasets to perform experiments, revealing the effect of the proposed method on vehicle image classification. The first one was the vehicle type image dataset (VTID) and the second was the vehicle make image dataset (VMID). Both datasets have been collected from the video surveillance system of Loei Rajabhat University in Loei province, Thailand.

1) Vehicle Type

A vehicle type is the description of the vehicle category that helps define the terms for classifying cars or other types of vehicles. In this research, we focused on five types of mainly personal vehicles in Thailand (Figure 33):

Figure 33. Five types of mainly personal vehicle in Thailand, (a) sedan, (b) hatchback, (c) sport utility vehicle (SUV), (d) pick-up, and (e) van

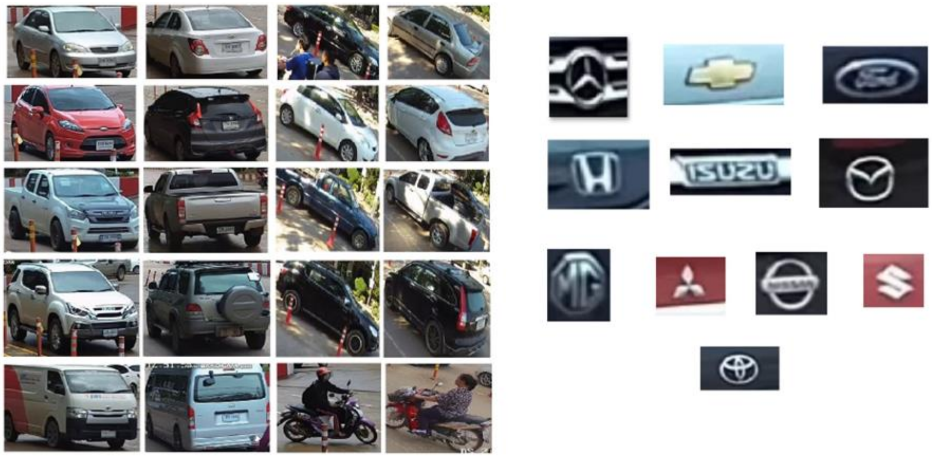
2) Vehicle Make

The vehicle's make is the brand of the vehicle and mostly the name of the company manufacturing the vehicle. People easily recognize the vehicle by seeing the logo because of its unique design and is familiar to most people (Figure 34). This can help a machine do the same thing. By locating and recognizing the vehicle logo, it is possible for a computer system to classify the vehicle make by analyzing the differences in each logo and figuring out how to categorize them.

Figure 34. Example of the vehicle logo in Thailand

The first dataset, VTID was a collection of five types of popular vehicles as described above. There were two versions of this dataset, VTID1 and VTID2. VTID1 was the smaller set consisting of 1,310 sample images. VTID2 was larger with a total of 4,356 images (Boonsirisumpun and Surinta, 2022). This chapter used only VTID2 for the experiment (Figure 35a).

The second dataset, VMID, was the collection of eleven vehicle logos in Thailand (Benz, Chevrolet, Ford, Honda, Isuzu, Mazda, MG, Mitsubishi, Nissan, Suzuki, and Toyota). The total number of images was 2,072 (Figure 35b).



(a) (b)

Figure 35. Example of (a) VTID2 and (b) VMID datasets

4.2.5 Experimental

The study of the effect of ensemble multiple CNNs methods with a partial training set for vehicle image classification in this research was separated into three steps. The first experiment was the performance of a single CNNs model on the full VTID2 and VMID datasets. The second was the ensemble of five CNNs models on the full VTID2 and VMID datasets. The last one was the ensemble of five CNNs models on partial VTID2 and VMID datasets.

1) Single CNNs model on full training set

The first experiment was designed to collect the initial performance of each CNNs on the dataset. By using every single CNNs model from the five chosen models (MobileNets V1, MobileNets V2, Inception V3, Inception V4, ResNet50) on the full-size training dataset with no ensemble technique. A 10-fold cross-validation was used to average the accuracy of the preprocessing. The experiments were run using Python 3.7.3 on an Intel Core i-7, 8th Gen, 4.0 GHz, Ram 8GB. The training method used the train from scratch for 50 epochs to compare with last year’s experiment (20 epochs). (Figure 36).

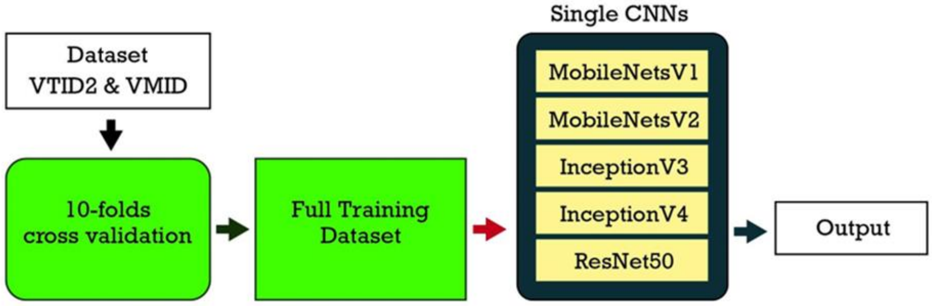


Figure 36. Processes of the first experiment

2) Ensemble multiple CNNs model on full training set

The second experiment was designed to study the effect of the ensemble technique on vehicle image classification. By using the same 10-fold cross-validation training dataset from the previous experiment, it replaced the single CNNs classifier using the ensemble of five CNNs models prediction together (MobileNets V1, MobileNets V2, Inception V3, Inception V4, and ResNet50). The combination of the output used both techniques from the ensemble method described in section 2, the unweighted average and majority vote. The experimental results recorded both the accuracy and time consumption to compare with the other experiments (Figure 37).

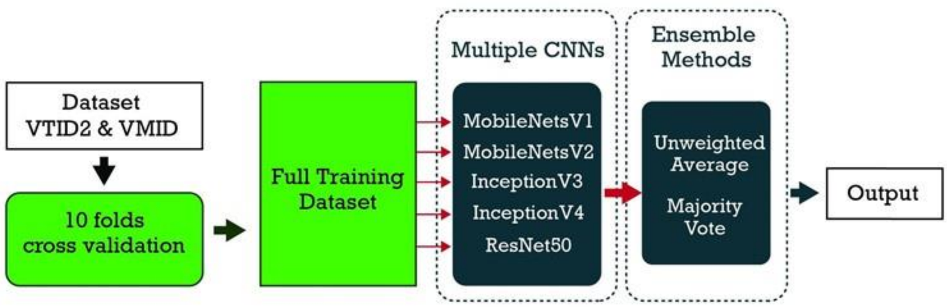


Figure 37. Processes of the second experiment

3) Ensemble multiple CNNs model on partial training set

The final experiment was designed to study the effect of the partial training dataset on the ensemble methods. By randomly selecting some sliced parts of the training set for each CNNs model. The size of the partial training set was chosen from three parameters (1/2, 1/3, and 1/5 of the full size). Five CNNs models from the previous experiments were also used. The unweighted average and majority vote were still performed. The experimental results recorded both the accuracy and time consumption as in the last experiment (Figure 38).

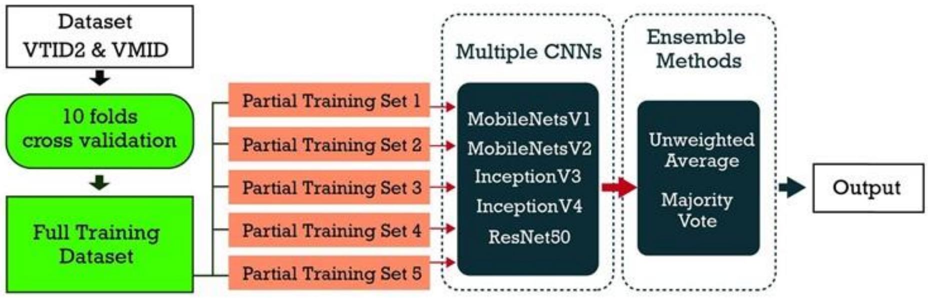


Figure 38. Processes of the third experiment

4.3 Results and Discussion

The results of the first experiment are shown in Table 10. These experiments were run using five CNNs models with two different epochs (20 and 50). The result showed that in 20 epochs, the performance of MobileNets V1 was close to that of Inception V3 with MobileNets V1 having the highest accuracy on VTID2 but Inception V3 was better on VMID. However, in 50 epochs, the inception V3 was overcome on both datasets. It could be concluded that the smaller model like MobileNets learned faster than the other when using fewer epochs. But with more epochs, the larger model could find the output better.

The training runtime of the first experiment is shown in Table 11. MobileNets V2 showed the advantages of the smallest model both in two datasets and a different number of epochs (20 and 50). The best result was on the VMID dataset, with only 18.57 minutes to finish the training process in 20 epochs. For VTID2, it seemed like the runtime was double, like the size of VTID2, which is twice the size of VMID. It could probably be concluded that the training time was increased following the ratio of increasing data and epochs.

Table 10 The accuracy of single CNNs method.

Network model/dataset	**VTID2 (20 epochs)**	**VTID2 (50 epochs)**	**VMID (20 epochs)**	**VMID (50 epochs)**
MobileNets V1	94.38	94.74	90.57	91.83
MobileNet V2	93.56	93.72	89.61	90.17
Inception V3	91.37	95.61	90.83	92.22
Inception V4	92.17	94.38	89.82	90.17
ResNet50	91.24	92.03	88.60	89.55

Table 11 The training runtime (min) of single CNNs method.

Network model/training time	**VTID2 (20 epochs)**	**VTID2 (50 epochs)**	**VMID (20 epochs)**	**VMID (50 epochs)**
MobileNets V1	50.72	128.32	22.14	63.96
MobileNet V2	42.18	106.54	18.57	52.63
Inception V3	65.87	167.78	32.15	84.70
Inception V4	81.54	208.10	43.24	112.56
ResNet50	74.21	191.23	36.52	103.28

The results of the second experiment are shown in Table 12. These experiments were challenged by the ensemble of five CNNs models in three different ways (ensemble of five MobileNet V1, ensemble of five Inception V3, and ensemble of the different five CNNs) using 50 epochs both in the unweighted average method and majority vote method. The results showed that the ensemble of the different five CNNs had the highest accuracy both in VTID2 and VMID, in which the unweighted was better than the majority vote. It could be concluded that the combination of different models could help to cover the weaknesses of other models better than using the same model five times.

Table 12 The accuracy of ensemble multiple CNNs method on full training set.

Ensemble multiple model/ dataset	Full VTID2 (50 epochs)		Full VMID (50 epochs)	
	Unweighted average	Majority vote	Unweighted average	Majority vote
5 MobileNets V1	95.33	94.98	92.37	91.83
5 Inception V3	96.01	95.93	92.89	92.54
5 models combination	96.15	95.89	93.11	92.89

The training runtime of the second experiment is shown in Table 13. The ensemble of five MobileNets V1 showed the fastest training speed in both of the two datasets. The five models in combination were slightly faster than five Inception V3 but, unfortunately, all three ensembles were much slower than the single CNNs methods. It could be concluded that the good accuracy of the ensemble method required a very long time in the training process.

Table 13 The training runtime (min) of ensemble multiple CNNs method on full training set.

Ensemble multiple model/training time	Full VTID2 (50 epochs)	Full VMID (50 epochs)
5 MobileNets V1	650.15	320.75
5 Inception V3	864.76	433.25
5 models combination	803.13	420.56

The results of the third experiment are shown in Table 14. These experiments were focused on the effect of the partial training set technique on the ensemble method. The results showed that when reducing the size of the training set, the accuracy was decreased but was still better than the single CNNs. The accuracy of the 1/2 and 1/3 partial training sets was higher than the highest score on the single CNNs model (Inception V3). It could be concluded that the decreasing of the training set was effect to the accuracy but the power of ensemble different CNNs models helped keep the total accuracy of the combination better than most single CNNs.

Table 14 The accuracy of ensemble multiple CNNs methods on the partial training set.

Ensemble multiple model/size of partial training set	Partial VTID2 (50 epochs)		Partial VMID (50 epochs)	
	Unweighted average	Majority vote	Unweighted average	Majority vote
1/2 partial+5 models combination	96.01	95.74	92.94	91.83
1/3 partial+5 models combination	95.85	95.93	92.03	91.65
1/5 partial+5 models combination	95.57	95.01	91.42	90.94

The training runtime of the third experiment is shown in Table 15. The effect of reducing the training data was obviously less time-consuming. The best size to choose for increasing the speed was a 1/5 partial training set. The overall time could be compared to every single CNNs method and the accuracy of partial training can still be acceptable and higher than single CNNs.

Table 15 The training runtime (min) of ensemble multiple CNNs methods on the full training set.

Ensemble multiple model+size of partial training set/ training time	Full VTID2 (50 epochs)	Full VMID (50 epochs)
1/2 partial+5 models combination	402.87	214.88
1/3 partial+5 models combination	276.17	142.67
1/5 partial+5 models combination	172.33	86.74

These experiments showed the effectiveness of ensemble multiple CNNs methods compared to a single model on vehicle type and vehicle make image recognition. By using five CNNs models that work together at the same time, the ensemble methods increased the recognition accuracy of both methods but compensated with exponential runtime. To fix the problem of time, the experiment was redesigned using a smaller piece (partial) of the training set instead, and achieved success in solving this time problem while keeping the high accuracy compared to using the full dataset.

4.3 Conclusion

In this chapter, the researchers proposed a new method, called ensemble multiple CNNs methods with partial training set, for vehicle image classification. By using the concept of ensemble method on a multiple CNNs model, and the idea of randomly slicing a small part of the training set to do a partial training instead of full training is able to reduce the runtime. The experimental results had satisfying outcomes. The ensembles of five CNNs models help to increase the accuracy of vehicle type and vehicle make (logo) image recognition. The precision of model prediction was improved in every combination and succeeded the previous model using only a single CNNs. The concept of the partial training set was suitable to solve the ensemble runtime problem. Slicing the part of the training set helped to decrease more than 60% of the completed ensemble process, but was also able to keep better accuracy than a single model.

For future work, this new method requires more experiments with other types of problems and a different dataset to ensure its performance. All hyper-parameters need to perform analysis. For example, the number of model combinations can probably be selected from the confidence value of each model instead of a fixed number. Additionally, the size of each partial training set can be weighted based on the performance of every single model instead of choosing the same size. A better fine-tuning parameter will help lead to better performance of the future method.

CHAPTER V

HYBRID GAN-YOLO: A SUPER-RESOLUTION MODEL FOR THAI LICENSE PLATE RECOGNITION

These are some problems to extract the information from Thai license plate character. One is the low quality of the image i.e., the noise or the camera performance. In general, Thai license plate has two rows of information. First row is a set of Thai alphabets follow by three or four Arabic numbers in large size. Second row is a Thai province name issued a plate in small size. However, some characters are difficult to identify with the low-resolution images. This research proposes a method to transform these images to a new higher resolution until there are capable to capture these characters call a "Hybrid GAN-YOLO". By combination three steps: Image Preprocessing, Super-Resolution generative adversarial network and YOLO network for object detection. Two image preprocessing (Warp Transform, Black&White), two GAN architectures (SRGAN and ESRGAN) and two YOLO networks (YOLOv5, YOLOv7) were used in this experiment. The experiment had performed with a real-environment Thai license plate image dataset call "Vehicle License Plate Dataset (VLID)" included both of high-quality license plate images and low-resolution images. The result showed that the hybrid of WarpTransform-ESRGAN-YOLOv7 outperformed others combination with the highest accuracy as 93.20%.

5.1 Introduction

License plate recognition is an important task in the field of computer vision and pattern recognition. It involves the identification and extraction of information from license plates in images or videos. License plate recognition systems are widely used for various applications such as traffic monitoring, toll collection, and law enforcement. However, the quality of license plate images can be a challenge for accurate recognition. This is particularly true for Thai license plates, which have two rows of information, including Thai alphabets, Arabic numbers, and a province name, with some characters difficult to identify with low-resolution images (See Figure 39).

Figure 39. Example of Thai license plate in low-resolution images.

To address these challenges, researchers have proposed various techniques such as feature extraction, pattern recognition, and deep learning-based methods. For example, Subhadhira using extreme learning machines (ELM) to conducted on Thai license plate images and the results showed an accuracy of 89.05% (Subhadhira et al., 2014), Kraisin and Kaothanthong proposed a method that combined histogram of oriented gradients (HOG) and extreme learning machine on a dataset of low-resolution province name images with the results achieved 90% accuracy (Kraisin & Kaothanthong, 2018), Rattanawong presented a Thailand license plate detection and recognition system that consists of two stages: license plate detection using the YOLOv3 model and license plate character recognition using a convolutional neural network (CNN) model. The experiment showed that their proposed system achieved an accuracy of 89.08% (Rattanawong et al., 2021). Thumthong proposed an automatic detection and recognition system for Thai vehicle license plates from CCTV images using the YOLOv4 object detection model for license plate detection and a CNN model for license plate character recognition. The results showed that the number recognition reach an accuracy of 94.20%, while the character recognition has an accuracy of 75.46% (Thumthong et al., 2021).

In this chapter, we propose a hybrid model that combines GAN and YOLO network to improve the recognition accuracy of Thai license plates with the addition of Image Preprocessing part. The proposed model, called Hybrid GAN-YOLO, aims to transform low-resolution images of Thai license plates to a new higher resolution until they are capable of capturing all characters. Specifically, two image preprocessings, two GAN architectures (SRGAN and ESRGAN), and two YOLO networks (YOLOv5 and YOLOv7), were used in the experiment. We evaluated the performance of the proposed method using a real-environment Thai license plate image dataset called “Vehicle License Plate Dataset (VLID)”, which included both high-quality license plate images and low-resolution images.

The results of the experiment showed that the hybrid of WarpTransform-ESRGAN-YOLOv7 outperformed other combinations with the highest accuracy of 93.20%. This indicates that the proposed Hybrid GAN-YOLO model is an effective approach for improving the recognition accuracy of Thai license plates. The rest of this chapter is organized as follows. In Section 5.2, we review related works in the field of license plate recognition. Section 5.3 describes the proposed Hybrid GAN-YOLO model in detail. Section 5.4 presents the experimental setup and evaluation matrices. Section 5.4 for results and discussion. And finally, Section 5.5 concludes the chapter and discusses future research directions.

5.2 Related Works

5.2.1 SRGAN

SRGAN, or Super-Resolution Generative Adversarial Network, is a type of deep neural network used for single-image super-resolution, which is the task of increasing the resolution of a low-resolution image to produce a high-resolution version. The network consists of two main components: a generator network and a discriminator network. The generator network takes a low-resolution image as input and tries to generate a high-resolution version of the same image that resembles the original high-resolution image. The discriminator network, on the other hand, takes a high-resolution image as input and tries to distinguish between the real high-resolution image and the generated high-resolution image produced by the generator network.

Both networks are trained together in an adversarial manner, where the generator tries to fool the discriminator into thinking that its generated high-resolution image is real, and the discriminator tries to correctly classify the real and generated high-resolution images. This results in a feedback loop, where the generator gets better at generating high-resolution images that look more like the original high-resolution images, and the discriminator gets better at distinguishing between real and generated high-resolution images (Ledig et al., 2017).

In this way, SRGANs are able to generate high-resolution images that are not only visually similar to the original high-resolution images but also contain rich details and textures. They have been widely used in various computer vision applications, such as image restoration, image synthesis, and others.

5.2.2 ESRGAN

ESRGAN, or Enhanced Super-Resolution Generative Adversarial Network, is a more advanced version of the SRGAN architecture. It is a type of deep neural network that aims to generate high-resolution images from low-resolution inputs with improved visual quality. ESRGAN builds upon the basic SRGAN architecture and incorporates several improvements to address some of the limitations of SRGAN. For example, ESRGAN uses a more complex generator network architecture that incorporates residual blocks and dense blocks, which help to capture fine details and textures in the high-resolution images. Additionally, ESRGAN uses a perceptual loss function that takes into account the high-level features of the high-resolution images, as well as the pixel-wise differences between the real and generated high-resolution images.

In terms of results, ESRGAN has been shown to produce high-resolution images with improved visual quality compared to SRGAN, including sharper edges, clearer textures, and more natural-looking results. It has been widely used in various computer vision applications, such as image super-resolution, image restoration, and others (Wang et al., 2018).

In general, ESRGAN is considered to be a more advanced and improved version of SRGAN, and it has been shown to produce high-resolution images with better visual quality compared to SRGAN. However, whether ESRGAN is better than SRGAN ultimately depends on the specific use case and the desired outcome. For some tasks, the improvements in visual quality provided by ESRGAN may not be necessary or relevant. In such cases, the simpler and more computationally efficient SRGAN may be a better choice. On the other hand, for tasks where high-quality and visually appealing results are important, ESRGAN may be the better choice.

5.2.3 YOLO v5

YOLO v5 is the fifth iteration of the YOLO architecture series developed by Ultralytics and released in 2020. This network model exhibits high detection accuracy and fast inference speed, achieving the fastest detection speed of up to 140 frames per second. Moreover, the YOLOv5 target detection network model boasts a small weight file size, which is approximately 90% smaller than the predecessors. This characteristic renders the YOLO v5 model suitable for deployment to embedded devices to facilitate real-time detection. Accordingly, the YOLO v5 network presents a host of advantages, such as its high detection accuracy, lightweight profile, and fast detection speed. (Jocher et al., 2021)

5.2.4 YOLO v7

Chien-Yao Wang released YOLOv7 in 2022. This network featuring enhanced detection speed and accuracy compared to its predecessors. Specifically, the network's architecture introduces E-ELAN, which utilizes expand, shuffle, merge cardinality to augment the network's learning ability continually while preserving the original gradient path. E-ELAN guides different groups of computational blocks to learn diverse features. Additionally, the network proposes a compound model scaling method that maintains the optimal structure and properties that the model had at its initial design.

In summary, the YOLOv7 network presents significant advancements in detection speed and accuracy through its novel E-ELAN architecture and compound model scaling method. Furthermore, the network optimization strategy introduces improvements in model re-parameterization and dynamic label assignment, enabling the network to achieve greater efficiency and effectiveness in object detection (Wang et al., 2023).

5.3 Hybrid GAN-YOLO for Thai License Plate Recognition

In this chapter, we propose a hybrid model that combines two deep learning architectures, GAN and YOLO, to address the problem of Thai vehicle license plate detection. The structure of this hybrid model can be seen in Figure 40.

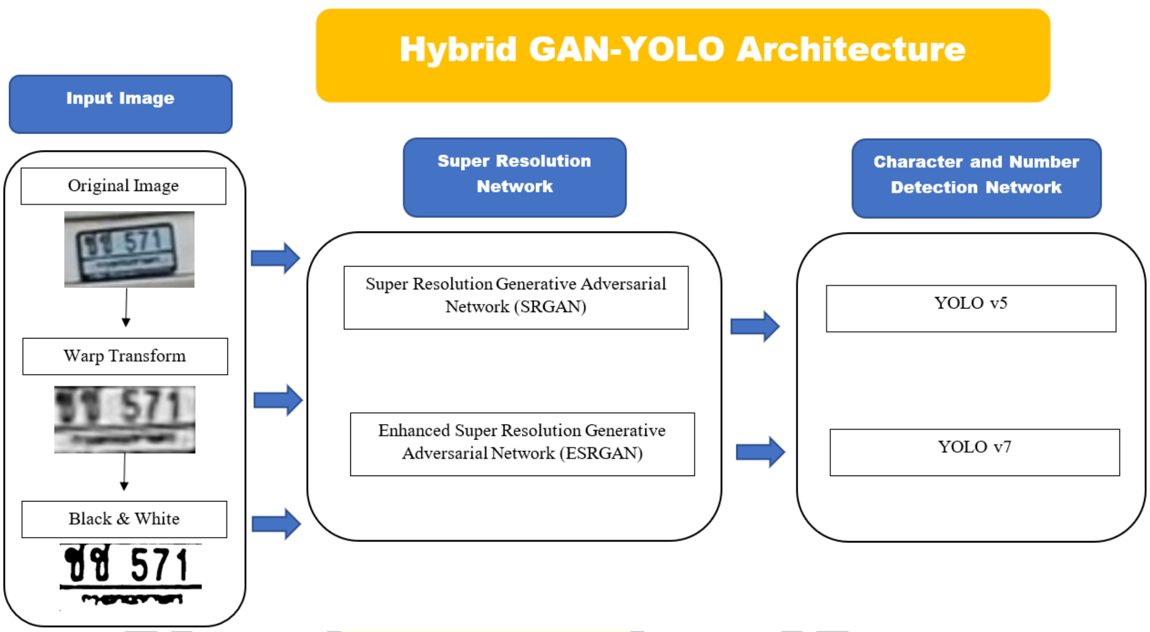


Figure 40. Structure of the hybrid GAN-YOLO.

The proposed hybrid structure begins with an input image, which is processed using an image preprocessing layer that includes warp and black-and-white transforms. The resulting image is then passed through a generative adversarial network (GAN) that uses deep neural networks to produce a high-resolution image. The high-resolution image is then passed through a YOLO architecture, which is used to detect the position of the characters and perform number and character recognition.

The use of GANs in image super-resolution has been shown to produce high-quality images that can improve the accuracy of object detection. By generating a high-resolution image, the proposed hybrid structure can improve the performance of the subsequent YOLO architecture in recognizing the characters on the license plate. The YOLO architecture is known for its ability to perform real-time object detection with high accuracy, which makes it well-suited for the license plate recognition problem.

The proposed hybrid GAN-YOLO structure is designed to take advantage of the benefits of both GAN and YOLO architectures. By combining these architectures, we aim to improve the accuracy of Thai license plate recognition. The warp and black-and-white transforms are used as image preprocessing techniques to improve the quality of the input image, while the GAN architecture is used to enhance the resolution of the image. The YOLO architecture is then used to accurately detect the position of the license plate characters and perform recognition. Overall, the proposed hybrid structure has the potential to improve the accuracy and efficiency of license plate recognition, which could have important applications in law enforcement and transportation management.

5.4 Experimental Setup and Evaluation Metrics

5.4.1 Dataset

To perform experiments in this study, the Thai Vehicle License plate Image Dataset (VLID) was used. The dataset was collected from the video surveillance system of Loei Rajabhat University in Loei province, Thailand. It contains two types of images, Hi-res and Lo-res, with different resolutions. The Hi-res type consists of 1,680 images that were used for training in the character detection process, while the Lo-res type consists of 1,991 images that were used for testing input in the super-resolution process.

The VLID was chosen as it is a suitable dataset for evaluating the performance of our proposed hybrid GAN-YOLO structure. The dataset contains a variety of Thai vehicle license plates with different sizes, colors, and font types, making it a challenging dataset for license plate recognition. By using this dataset, we aim to demonstrate the effectiveness of our hybrid model in accurately detecting the license plate characters.

The Hi-res images were used to train the character detection process, as they have a higher resolution that allows for more accurate character localization. On the other hand, the Lo-res images were used as input for the super-resolution process, as they have a lower resolution that requires the use of a super-resolution algorithm to increase the image quality (See Figure 41). By using these two types of images, we aim to evaluate the performance of our hybrid model in both the character detection and super-resolution processes.

a) b)

Figure 41. Example images of VLID. a) Hi-res, b) Lo-res.

ก	A1	ฐ	A16	ฟ	A31
ข	A2	ฑ	A17	ภ	A32
ฃ	A3	ฒ	A18	ม	A33
ค	A4	ณ	A19	ย	A34
ฅ	A5	ด	A20	ร	A35
ฆ	A6	ต	A21	ล	A36
ง	A7	ถ	A22	ว	A37
จ	A8	ท	A23	ศ	A38
ฉ	A9	ธ	A24	ษ	A39
ช	A10	น	A25	ส	A40
ซ	A11	บ	A26	ห	A41
ฌ	A12	ป	A27	ฬ	A42
ญ	A13	ผ	A28	อ	A43
ฎ	A14	ฝ	A29	ฮ	A44
ฏ	A15	พ	A30		

Figure 42. Thai alphabet Labeling.

For the class labels, this research labeled the group of Thai character in vehicle license plate into 3 parts. First, there are 44 Thai alphabet characters were labeled for class A1 – A44

(See Figure 42). Second, 32 Thai province names that found in the license plate dataset were labels for abbreviation version of their name (See Figure 43). Finally, the 10 classes of Arabic number were labels in class N0 – N9 (See Figure 44).

กรุงเทพ	BKK	จันทบุรี	CHAN	ร้อยเอ็ด	RE
ชัยภูมิ	CYP	ชลบุรี	CHON	ศรีษะเกษ	SSK
ขอนแก่น	KHON	ยโสธร	YST	ตาก	TAK
นครราชสีมา	KOR	กาฬสินธุ์	KAL	อยุธยา	NSA
เลย	LOE	เชียงใหม่	CHM	นครปฐม	NPT
อุดรธานี	UDON	เชียงราย	CHR	อุทัยธานี	UTT
บุรีรัมย์	BRR	พะเยา	PY	อำนาจเจริญ	ANC
สุรินทร์	SUR	กำแพงเพชร	KPP	ลำพูน	LP
อุบลราชธานี	UBON	แม่ฮ่องสอน	MHS	ลำปาง	LAMP
มหาสารคาม	MHK	ลพบุรี	LB	ชุมพร	CHP
สุพรรณบุรี	SUPB	สุราษฎร์ธานี	SRTN		

Figure 43. Thai province name Labeling.

0	N0	5	N5
1	N1	6	N6
2	N2	7	N7
3	N3	8	N8
4	N4	9	N9

Figure 44. Arabic number Labeling.

In summary, the Thai Vehicle License plate Image Dataset (VLID) was used to evaluate the performance of our proposed hybrid GAN-YOLO structure. The dataset contains two types of images, Hi-res and Lo-res, which were used for training and testing in the character detection and super-resolution processes, respectively. The use of this dataset allowed us to demonstrate the effectiveness of our hybrid model in accurately detecting license plate characters and improving the image quality of low-resolution images.

5.4.2 GAN-YOLO Model Selection

The model selection for our proposed technique involved the consideration of two Generative Adversarial Network (GAN) architectures for the super-resolution process: SRGAN and ESRGAN, and two You Only Look Once (YOLO) architectures, YOLOv5 and YOLOv7, for the character detection process. In total, our proposed technique includes four different combinations of GAN-YOLO models: SRGAN-YOLOv5, SRGAN-YOLOv7, ESRGAN-YOLOv5, and ESRGAN-YOLOv7, which were evaluated using the Thai Vehicle License Plate Image Dataset (VLID) as described in previous section. The selection of GAN-YOLO models was made based on their performance in the two individual processes as well as their combination performance.

5.4.2 Evaluation Metrices

In this research, we evaluated the performance of our proposed hybrid GAN-YOLO models using two commonly used evaluation metrics: Accuracy and Mean Average Precision (mAP). Accuracy is a standard metric for measuring the correctness of classification models, which indicates the percentage of correctly classified samples out of the total samples. In the context of our problem, it measures the percentage of correctly detected license plates.

On the other hand, mAP is a widely used metric for evaluating object detection models, which measures the accuracy and precision of the model in detecting and localizing objects in the image. In particular, mAP computes the area under the precision-recall curve, which is a plot of the true positive rate against the false positive rate at various thresholds. A higher mAP score indicates better performance in detecting and localizing objects.

To evaluate the performance of the proposed models, each model was trained and tested on the Thai vehicle license plate image dataset (VLID), and their accuracy and mean average precision (mAP) scores were measured. Various combinations of GAN-YOLO models, including SRGAN-YOLOv5, SRGAN-YOLOv7, ESRGAN-YOLOv5, and ESRGAN-YOLOv7, were compared in terms of their effectiveness. The experiments were conducted in Python 3.8 using the TensorFlow and Keras libraries, running on a Google Colab GPUs. The experimental results provide valuable insights into the effectiveness of each combination of GAN-YOLO models in addressing the problem of Thai license plate recognition.

5.5 Results and Discussion

In this section, we present the results of our proposed hybrid GAN-YOLO models using different combinations of GAN and YOLO architectures on the Thai vehicle license plate recognition problem. We evaluated the performance of our models based on two widely used evaluation metrics, accuracy, and mean average precision (mAP).

As shown in table 16, we achieved high accuracy and mAP for all of our proposed models, but the results varied depending on the combination of GAN and YOLO architecture used. Among the four different combinations, the WarpTransform-ESRGAN-YOLOv7 model achieved the best results, with an accuracy of 93.20% and a mAP of 0.589.

Table 16 The evaluation metrices of the Hybrid GAN-YOLO.

Image Preprocessing	***GAN-YOLO Model Combination***	***Accuracy***	***mAP***
Original Image	SRGAN-YOLOv5	89.98	0.561
	SRGAN-YOLOv7	91.90	0.581
	ESRGAN-YOLOv5	91.54	0.576
	ESRGAN-YOLOv7	92.61	0.587
Original Image + Warp Transforming	SRGAN-YOLOv5	90.39	0.561
	SRGAN-YOLOv7	91.37	0.575
	ESRGAN-YOLOv5	92.11	0.582
	ESRGAN-YOLOv7	***93.20***	***0.589***
Original Image + Warp Transforming + Black & White	SRGAN-YOLOv5	89.38	0.558
	SRGAN-YOLOv7	90.84	0.569
	ESRGAN-YOLOv5	91.06	0.573
	ESRGAN-YOLOv7	90.93	0.569

Figure 45 illustrates a selection of output images obtained from processing a low-resolution input image through an image preprocessing stage consisting of both warp transformation and black and white conversion, followed by super resolution processing via a GAN network. These images were subsequently subjected to character detection using the Yolo network. Our analysis of the results indicates that the use of warp transformation preprocessing leads to improved performance of the GAN-YOLO system. However, we observed that applying the black and white preprocessing step resulted in the removal of data from the last line of the license plate, specifically the province name.

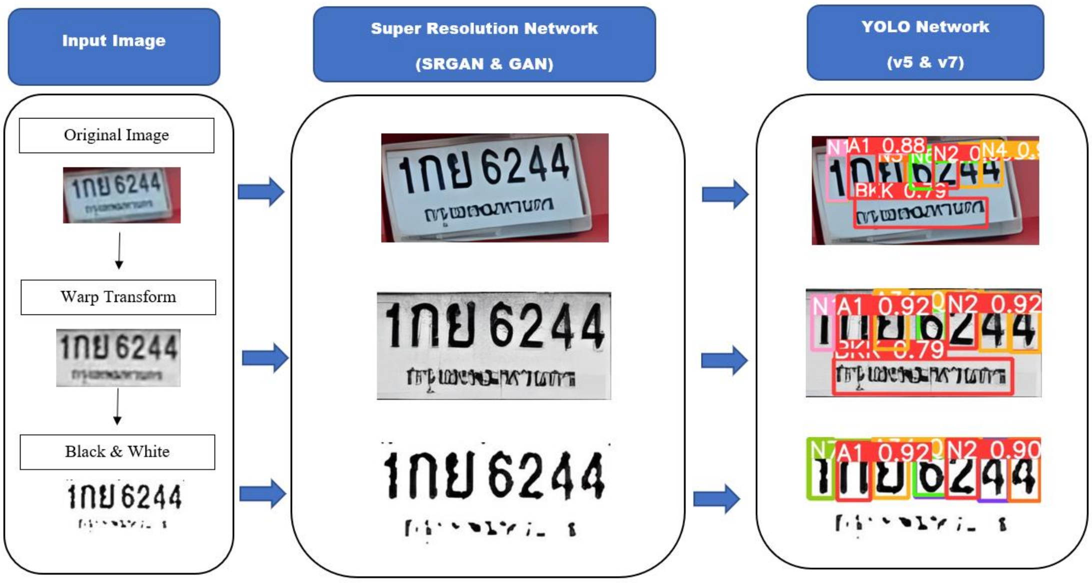


Figure 45. Result of Hybrid GAN-YOLO.

The results suggest that the use of Warp Transform with ESRGAN and YOLOv7 architectures in combination provides the best solution for the Thai vehicle license plate recognition problem. The ESRGAN is a state-of-the-art GAN architecture that is well-suited for super-resolution tasks, and YOLOv7 is an advanced version of the YOLO architecture that achieves high accuracy and detection speed.

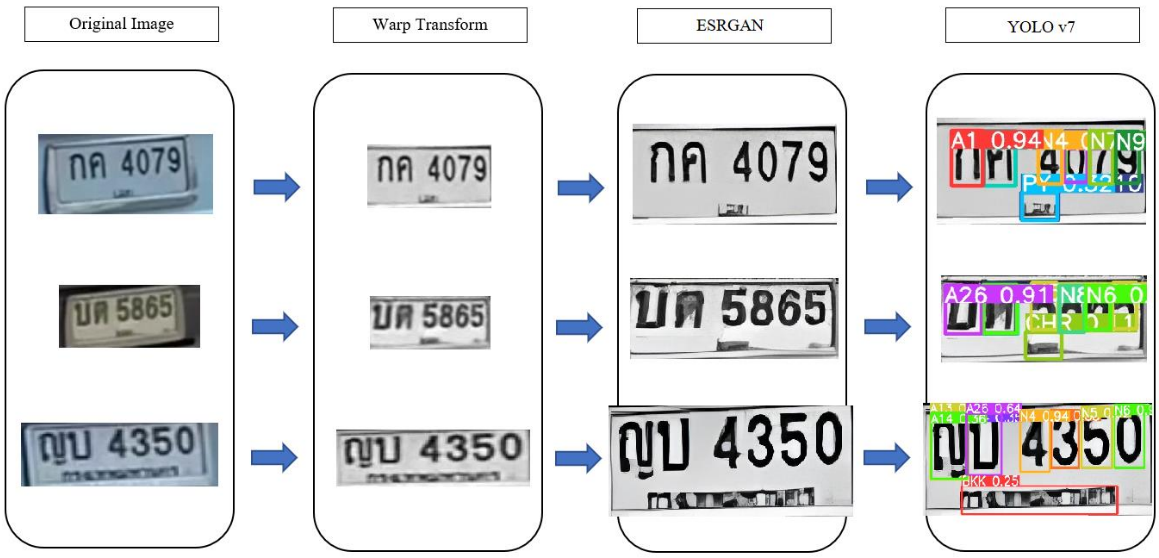


Figure 46. Example of Warp-ESRGAN-YOLOv7 results.

Figure 46 depicts the outcomes obtained from WarpTransform-ESRGAN-YOLOv7. The experimental findings indicate that this combination can effectively enhance the visual quality of the original characters present in the image, as seen in both the first and second rows. In the case of the first row, this technique demonstrates significant improvement in the sharpness of each character. However, for the second row, although the quality of the entire word is improved, YOLOv7 may not be able to accurately identify the province name.

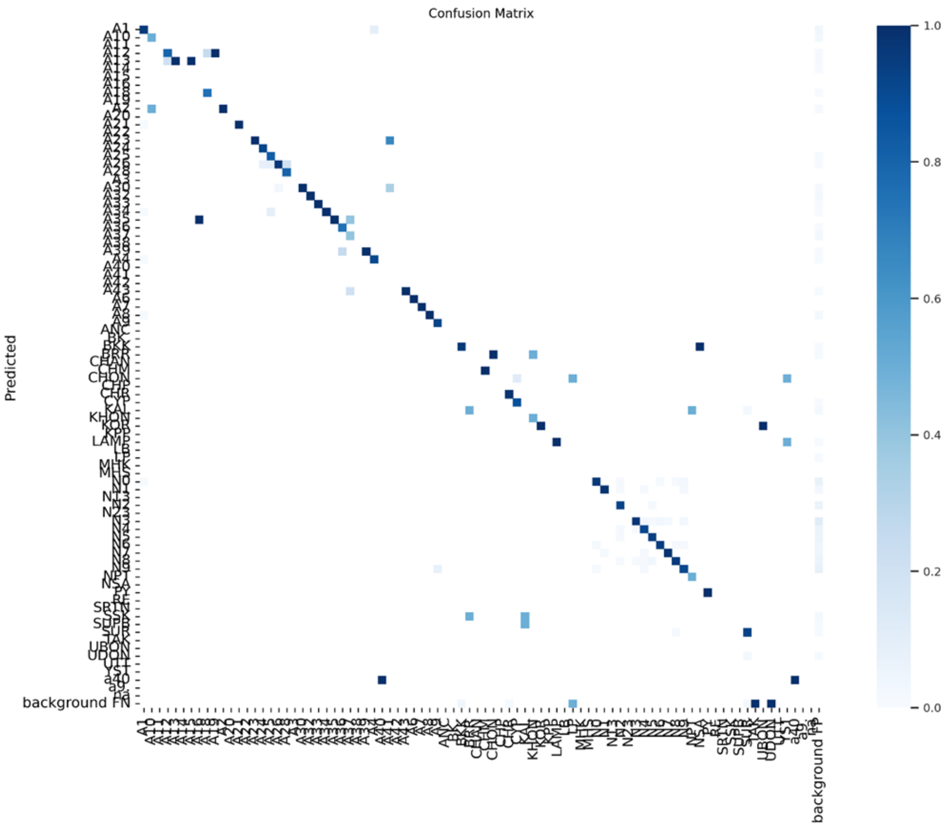


Figure 47. Result of Hybrid GAN-YOLO.

The confusion matrix presented in Figure 47 illustrates the performance of the character recognition model. The results indicate that the model performs well in recognizing Thai alphabets and numbers, but it performs slightly worse in recognizing province names. The analysis shows a high accuracy score for approximately 20-25 Thai alphabets, while some of the alphabets score lower, and about 10 alphabets are not represented in the training dataset.

However, for province names, the model's accuracy score is lower for approximately 10 out of 32 province names present in the dataset. This could be due to the longer word length of province names as compared to individual alphabets, as well as their smaller size in the training dataset.

In summary, the results of our experiments demonstrate the effectiveness of the proposed hybrid GAN-YOLO models in addressing the Thai vehicle license plate recognition problem. The

WarpTransform-ESRGAN-YOLOv7 model outperformed the other models in terms of accuracy and mAP, suggesting that this combination is a promising solution for license plate recognition tasks.

5.6 Conclusion

In conclusion, we have proposed a hybrid deep learning model to tackle the problem of Thai vehicle license plate detection. We have demonstrated the effectiveness of our proposed approach by comparing four different combinations of GAN and YOLO models. Our experiments were conducted on the Thai Vehicle License Plate Image Dataset (VLID), collected from the video surveillance system of Loei Rajabhat University in Loei province, Thailand.

Our experimental results show that the WarpTransform-ESRGAN-YOLOv7 combination outperformed the other three combinations in terms of both accuracy and mAP. The model achieved an accuracy of 93.20% and mAP of 0.589, demonstrating its robustness in handling different license plate types and conditions.

In summary, our proposed hybrid deep learning model shows promising results in the task of Thai vehicle license plate detection. Our approach can be applied to other similar object detection problems and can be extended to support multi-lingual license plates. We hope that our work will inspire further research in this field, leading to even more accurate and robust license plate recognition systems.

CHAPTER VI

CONCLUSIONS

The purpose of this dissertation was to develop an automatic vehicle detection and classification system using advanced deep learning architectures. The study focused on solving the problem of vehicle detection and classification using state-of-the-art deep learning methods. The dissertation was divided into three parts, each addressing a specific aspect of the problem. In Part I, advanced convolutional neural networks were used to classify five types of Thai vehicles. Part II, ensemble methods were used to combine multiple CNN models to classify both vehicle types and vehicle makes (logos). Finally, Part III, a hybrid GAN-YOLO model was developed to recognize Thai character on the vehicle license plates. The results of the study showed that the proposed models outperformed existing methods in terms of accuracy, speed, and size.

Part I of the dissertation focused on using advanced convolutional neural networks to classify five types of vehicles. The study compared nine different CNN models with data augmentation to find the best model for the task. The results showed that MobileNets outperformed the other architectures in terms of accuracy, speed, and size. The use of data augmentation helped to improve the performance of the models. The findings suggest that MobileNets is a suitable method for vehicle type recognition on all computing platforms, including mobile devices.

Part II of the dissertation focused on using ensemble methods to combine multiple CNN models to classify both vehicle types and logos. The study proposed a new technique for random choosing the training data called "A Partial Training Set." The results showed that using a combination of five different CNNs with a partial training dataset improved the accuracy of vehicle type and logo recognition. The use of ensemble methods helped to increase the performance of the model and also help reducing the overall runtime of the process.

Finally, part III focused on creating a hybrid GAN-YOLO model for license plate recognition. The study used two image preprocessing techniques, two different GAN architectures and two different YOLO networks to find the best combination. The results showed that the hybrid of WarpTransform-ESRGAN-YOLOv7 outperformed the other combinations in terms of accuracy. The use of a GAN for super-resolution helped to improve the quality of low-resolution testing images before recognition.

In conclusion, this dissertation presented an Automatic Vehicle Detection and Classification System using Advanced Deep Learning Architectures to address the challenges of vehicle detection and classification. The three main parts of the dissertation focused on vehicle type recognition, vehicle make and logo recognition, and license plate recognition, respectively. Through the use of advanced deep learning techniques, such as convolutional neural networks (CNNs), ensemble methods, and generative adversarial networks (GANs), we were able to achieve accurate and efficient results.

The overall contributions of this dissertation are significant. The study provides a comprehensive approach to solving the problem of vehicle detection and classification using advanced deep learning architectures. The proposed models outperform existing methods in terms of accuracy, speed, and size. The use of ensemble methods and partial training datasets helped to improve the performance of the models and reduce the overall runtime of the process. The findings have implications for real-world applications, including traffic monitoring and surveillance systems.

Despite the significant contributions of this study, there are some limitations that should be addressed in future research. One limitation is that the proposed models were tested on a limited set of datasets. Future studies should consider using more diverse datasets to test the models' performance. Additionally, the proposed models may not generalize well to other geographic locations. Future studies should consider developing models that can be adapted.

REFERENCES

Alsultanny, Y. A., & Aqel, M. M. (2003). Pattern recognition using multilayer neural-genetic algorithm. *Neurocomputing*, *51*, 237-247.

Bautista, C. M., Dy, C. A., Mañalac, M. I., Orbe, R. A., & Cordel, M. (2016). Convolutional neural network for vehicle detection in low resolution traffic videos. 2016 IEEE Region 10 Symposium (TENSYMP),

Bishop, C. M. (1995). *Neural networks for pattern recognition*. Oxford university press.

Boonsirisumpun, N., & Puarungroj, W. (2018). Loei fabric weaving pattern recognition using deep neural network. 2018 15th International joint conference on computer science and software engineering (JCSSE),

Chen, R.-C. (2019). Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning. *Image and Vision Computing*, *87*, 47-56.

Chollet, F. (2016). Building powerful image classification models using very little data. *Keras Blog*, *5*, 90-95.

Clady, X., Negri, P., Milgram, M., & Poulenard, R. (2008). Multi-class vehicle type recognition system. Artificial Neural Networks in Pattern Recognition: Third IAPR Workshop, ANNPR 2008 Paris, France, July 2-4, 2008 Proceedings 3,

Dechter, R. (1986). Learning while searching in constraint-satisfaction problems.

Dietterich, T. G. (2000). Ensemble methods in machine learning. Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1,

Dogan, A., & Birant, D. (2019). A weighted majority voting ensemble approach for classification. 2019 4th International Conference on Computer Science and Engineering (UBMK),

Dong, Z., Wu, Y., Pei, M., & Jia, Y. (2015). Vehicle type classification using a semisupervised convolutional neural network. *IEEE transactions on intelligent transportation systems*, *16*(4), 2247-2256.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2015). Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *38*(1), 142-158.

Goh, Z. L. (2021). *Model Optimization for Vehicle Recognition on Edge Devices* Monterey, CA;

Naval Postgraduate School].

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets, in 'Advances in Neural Information Processing Systems 27', Curran Associates. In: Inc.

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

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huttunen, H., Yancheshmeh, F. S., & Chen, K. (2016). Car type recognition with deep neural networks. 2016 IEEE intelligent vehicles symposium (IV),

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. International conference on machine learning,

Jocher, G., Stoken, A., Borovec, J., Chaurasia, A., Changyu, L., Hogan, A., Hajek, J., Diaconu, L., Kwon, Y., & Defretin, Y. (2021). ultralytics/yolov5: v5. 0-YOLOv5-P6 1280 models, AWS, Supervise. ly and YouTube integrations. *Zenodo*.

Kraisin, S., & Kaothanthong, N. (2018). Accuracy improvement of a province name recognition on Thai license plate. 2018 International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP),

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.

Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., & Matas, J. (2018). Deblurgan: Blind motion deblurring using conditional adversarial networks. Proceedings of the IEEE conference on computer vision and pattern recognition,

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, *3361*(10), 1995.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, *1*(4), 541-551.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278-2324.

Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., & Wang, Z. (2017). Photo-realistic single image super-resolution using a generative adversarial network. Proceedings of the IEEE conference on computer vision and pattern recognition,

Li, J., Zhao, W., & Guo, H. (2009). Vehicle type recognition based on Harris corner detector. International Conference on Transportation Engineering 2009,

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14,

Ma, Z., Chang, D., Xie, J., Ding, Y., Wen, S., Li, X., Si, Z., & Guo, J. (2019). Fine-grained vehicle classification with channel max pooling modified CNNs. *IEEE Transactions on Vehicular Technology*, *68*(4), 3224-3233.

Maikaew, P. (2018, 3 January 2018). Headwinds expected for auto industry. *Bangkok Post*.

Mikołajczyk, A., & Grochowski, M. (2018). Data augmentation for improving deep learning in image classification problem. 2018 international interdisciplinary PhD workshop (IIPhDW),

Mori, S., Nishida, H., & Yamada, H. (1999). *Optical character recognition*. John Wiley & Sons, Inc.

Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of electronic imaging*, *16*(4), 049901.

Neubeck, A., & Van Gool, L. (2006). Efficient non-maximum suppression. 18th international conference on pattern recognition (ICPR'06),

Pal, S. K., & Wang, P. P. (1996). *Genetic algorithms for pattern recognition*. CRC press.

Polikar, R. (2012). Ensemble Learning in Ensemble Machine Learning: Methods and Applications; Zhang, C., Ma, Y., Eds. In: Springer: Berlin/Heidelberg, Germany.

Puarungroj, W., & Boonsirisumpun, N. (2018). Thai license plate recognition based on deep learning. *Procedia Computer Science*, *135*, 214-221.

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Rattanawong, S., Hsu, G.-S., & Chung, S.-L. (2021). Thailand license plate detection and recognition. 2021 25th International Computer Science and Engineering Conference (ICSEC),

Raza, K. (2019). Improving the prediction accuracy of heart disease with ensemble learning and majority voting rule. In *U-Healthcare Monitoring Systems* (pp. 179-196). Elsevier.

Re, M., & Valentini, G. (2012). Ensemble methods: a review in Advances in Machine Learning and Data Mining for Astronomy (ed. Kumar, V.) 563–594. In: Chapman & Hall.

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

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

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

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. Proceedings of the IEEE conference on computer vision and pattern recognition,

Sawicki, D. S. (1993). *Traffic radar handbook: A comprehensive guide to speed measuring systems*. Grove Enterprises Brasstown, NC, USA.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Sewell, M. (2011). Ensemble methods. *Relatório Técnico RN/11/02, University College London Departament of Computer Science*.

Sharif Razavian, A., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN features off-the-shelf: an astounding baseline for recognition. Proceedings of the IEEE conference on computer vision and pattern recognition workshops,

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of big data*, *6*(1), 1-48.

Sifre, L., & Mallat, P. S. (2014). Rigid-motion scattering for image classification author. *English. Supervisor: Prof. Stéphane Mallat. Ph. D. Thesis. Ecole Polytechnique*, *2*.

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

Špaňhel, J., Sochor, J., & Makarov, A. (2018). Vehicle fine-grained recognition based on convolutional neural networks for real-world applications. 2018 14th symposium on neural networks and applications (NEUREL),

Subhadhira, S., Juithonglang, U., Sakulkoo, P., & Horata, P. (2014). License plate recognition application using extreme learning machines. 2014 Third ICT International Student Project Conference (ICT-ISPC),

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. Proceedings of the AAAI conference on artificial intelligence,

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition,

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. Proceedings of the IEEE conference on computer vision and pattern recognition,

Thomas, A., Harikrishnan, P., Palanisamy, P., & Gopi, V. P. (2020). Moving vehicle candidate recognition and classification using inception-resnet-v2. 2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC),

Thumthong, W., Meesud, P., & Jarupunphol, P. (2021). Automatic detection and recognition of Thai vehicle license plate from CCTV images. 2021 13th International Conference on Information Technology and Electrical Engineering (ICITEE),

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

Vapnik, V. (1999). *The nature of statistical learning theory*. Springer science & business media.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. Proceedings of the IEEE/CVF

Conference on Computer Vision and Pattern Recognition,

Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Change Loy, C. (2018). Esrgan: Enhanced super-resolution generative adversarial networks. Proceedings of the European conference on computer vision (ECCV) workshops,

Wu, Y., Ianakiev, K., & Govindaraju, V. (2002). Improved k-nearest neighbor classification. *Pattern recognition*, *35*(10), 2311-2318.

Yoo, B., Choi, Y., & Choi, H. (2018). Fast depthwise separable convolution for embedded systems. Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part VII 25,

Yu, S., Wu, Y., Li, W., Song, Z., & Zeng, W. (2017). A model for fine-grained vehicle classification based on deep learning. *Neurocomputing*, *257*, 97-103.

Zhang, B. (2012). Reliable classification of vehicle types based on cascade classifier ensembles. *IEEE transactions on intelligent transportation systems*, *14*(1), 322-332.

Zhang, H., Berg, A. C., Maire, M., & Malik, J. (2006). SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06),

Zhou, Y., & Cheung, N.-M. (2016). Vehicle classification using transferable deep neural network features. *arXiv preprint arXiv:1601.01145*.

BIOGRAPHY

NAME	Narong Boonsirisumpun
DATE OF BIRTH	09/02/2524
PLACE OF BIRTH	Bangkok
ADDRESS	234 Loei-Chiangkhan Rd. Faculty of Science, Loei Rajabhat University
POSITION	Lecturer
PLACE OF WORK	Loei Rajabhat University
EDUCATION	2001 Bachelor of Science (B.Sc.) Computer Science, Kasetsart University 2004 Master of Science (M.Sc.) Computer Science, Chulalongkorn University 2023 Doctor of Philosophy (Ph.D.) Information Technology, Mahasarakham University