VEHICLE CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORK

WATCHARIN MAUNGMAI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTATIONAL INTELLIGENCE SYSTEM

INTERNATIONAL COLLEGE

KING MONGKUTS INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

KMITL-2018-IC-M-011-05

VEHICLE CLASSIFICATION WITH DEEP CONVOLUTIONAL NEURAL NETWORK

WATCHARIN MAUNGMAI

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF ENGINEERING IN COMPUTATIONAL INTELLIGENCE SYSTEM

INTERNATIONAL COLLEGE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2018

KMITL-2018-IC-M-011-05

Thesis Title	Vehicle Classification with Deep Convolutional Neural Network
Student	Watcharin Maungmai
Student ID	60610017
Degree	Master of Engineering
Program	Computational Intelligence System
Thesis Advisor	Asst.Prof.Dr Chaiwat Nuthong

ABSTRACT

Nowadays, there are many traffic surveillance systems which are installed in almost every city to record events and traffic. The surveillance system is used for various objectives, e.g. vehicles searching and real-time traffic monitoring, etc. For the searching purpose, the system can be used by policeman such as outlaws vehicle identification in crime. Typically, the officers manually identify the vehicle in recorded video according to its appearances. Although the accuracy of this approach is good, it is time-consuming and inclined to faults due to human fatigue for long duration videos. Moreover, hiring employees is costly. Recently, there are several machine learning methods which can be applied to classify vehicles, e.g. Fuzzy Logic, Decision Tree, Adaboost, Random Forest, Neural Network, etc. Convolutional Neural Network (CNN) is also one of such methods. CNN is a type of Deep Learning which is in the category of the neural network. The method is very well-known in image recognition field at the present because of its performance.

In this research, CNN is chosen to be a proposed method or a classifier for vehicle classification. In these days, there are several paper which applied CNN with vehicle classification and those CNNs performed very well in vehicle classification. Most papers achieved more than 80 percent accuracy. In the proposed method, there are two techniques which could help to improve the performance of the CNN model i.e., pretrained weight and data augmentation. These two techniques are set as parameters in the experiment in order to measure the improvement. In the proposed vehicle classification, there are two vehicle characteristics, i.e. types and colors. Types consist of four classes while colors consist of seven classes. CNN is then used as to classify vehicle images. The experimental results show that CNN can achieve high performance in real-world applications.

ACKNOWLEDGEMENTS

I would like to thank Asst. Prof.Dr.Chatwat Nuthong, my advisor, for providing me various resources, commendation, guidance, and knowledge. Furthermore, I would like to thank Dr.Ukrit Watchareeruetai for giving us suggestion and criticism during the research. Special thanks to International College, King Mongkuts Institute of Technology Ladkrabang for providing us resources and supports.

Bangkok, June 2019

Watcharin Maungmai

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Nowadays, surveillance cameras are installed almost everywhere in cities. The main objectives of installing surveillance systems are real-time monitoring and events searching. In this paper, the authors focus only on events searching. For the searching objective, the surveillance system can be used by police officers. For example, in order to search for specific vehicle. In general, the officers require the information of vehicles characteristics, e.g. vehicles color, vehicles type as a clue for vehicle identification. The officers often spend a lot of time monitoring recorded videos by themselves. Typically, searching time is usually more than video duration and they have to repeat the searching task again several times. In addition, the officers might make some mistakes with their weariness after a period of searching. In order to solve such problems, vehicle classification can be utilized in order to assist the vehicle searching. Various methods are applied in vehicle classification at present. In this research, the proposed system aims to achieve high performance of vehicle classification. The system could be an important component in automatic vehicle searching system which could eliminate human involvement in order to reduce searching time and cost. The experiment results show that the proposed system achieve high accuracy rate in vehicle classification which is high enough to be used in real world.

1.1 Problem Descriptions

The major objectives of traffic surveillance system are real-time traffic monitoring, object searching, etc. For the searching purpose, if police officer wants to search for a bad person's vehicle, the officer requires some information of vehicle characteristics, e.g. color, type, etc as a clue for vehicle identification. The manually searching progress spends a lot of time monitoring recorded videos. And if the video has very long duration. it might make some error or inaccurate because of human fatigue. In order to solve this problem, vehicle classification can be utilized to assist the vehicle searching.

1.2 Research Objectives

In order to indicate the successfulness of the project, several goals have been set. The following list shows all the goals which needed to be satisfied:

- To study deep learning (Convolutional Neural Network)

- To apply deep learning with vehicle type and vehicle color classification
- To improve the performance of the classification to be better than previous works
- The overall system's accuracy is expected to be more than 85 percent.

1.3 Proposed System

According to the problems, this research proposes the classification systems which are designed and constructed based on Convolutional Neural Network. These systems are expected to achieve all objectives which are mentioned in the previous section. The systems requires one input that is an vehicle image. The predicted class of each vehicle image is the output of this system.

1.4 Scope of the Study

In this research, the proposed system is constructed in order to achieve high performance of vehicle classifications. In addition, the proposed system consists of two classification models which are vehicle color classifier and vehicle type classifier. The input of this system are vehicle images. Saripan et al. [1]'s vehicle image dataset is used in this research. The vehicle images are extracted from a video systematically by Saripan's system. The system is not perfect as human, there are some overlapped vehicles images as error which are included in the dataset. The output classes of our proposed system are the same as Saripan's work in order to measure and improve the classification performance. There are seven classes in the vehicle color classification, i.e. Black, White, Blue, Green, Yellow, Red and Unknown. In vehicle type classification, there are four classes which are Small, Medium, Large and Unknown.

1.5 Thesis Structure

This thesis contains seven chapters. Chapter 2 discusses the concerning literature review and shows the concepts related to this project. Chapter 3 contains the background knowledge of Deep learning and Convolutional Neural Network. The detail explanation of the proposed system is shown in chapter 4. Chapter 5 presents the experimental setup and results. Lastly, there are conclusion and discussion which are mentioned in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

This chapter is devoted for literature review, which includes past studies, related works and theories that involve in this research. Detailed information can be found in following sections.

2.1 Tree-based vehicle classification system

Saripan et al. [2] proposed a tree-based vehicle classification system which required a surveillance video and vehicles characteristics as inputs. This work is proposed based on search system. The system consists of three modules, i.e. feature extraction, classification, and search manager. Figure 2.1 shows the overview of the system. The video is used as an input at the beginning in feature extraction module. The module crops vehicle images from the video systematically and extracts the features of those images which are listed in the figure 2.2. These extracted feature data is then sent to the classification module. Figure 2.3 shows the overview of the classification module. There are two classification, i.e. type and color. In type classification, four classes are categorized, i.e. small, medium, large and unknown. There are seven classes in color classification, i.e. black, white, blue, green, yellow, red, and unknown. Note that, both classifications consist of unknown class. This class contains the vehicle with ambiguous characteristics and irrelevant colors, e.g. overlapped vehicles, brown color, etc. Overall possible target classes in type and color classification are shown in figure 2.4. The results from classification module are sent to the search manager module. This module then stores and filters the results according to the given query commands. Figure 2.5 shows the experiment result of the classification. In type classification, random forest achieved the highest accuracy which is 79.82%. The best predictor of color classifier from Saripan's experiment is adaboost with 69.29% accuracy.

2.2 Vehicle Color Recognition using Convolutional Neural Network

Rachmadi et al. [3] proposed a CNN framework in order to recognize vehicle's color. Typically, Convolutional neural network is designed to solve classification problem based on shape information. The authors wanted to prove that Convolutional neural network can be able to do the classification based on color distribution as well. They proposed the CNN structure which can classify color in 8 classes.In detail, the authors implement the CNN by using Caffe framework and publicly vehicle color recognition dataset which contains 15,601 vehicle images with 8 classes. The CNN structure is shown in Figure 2.6.

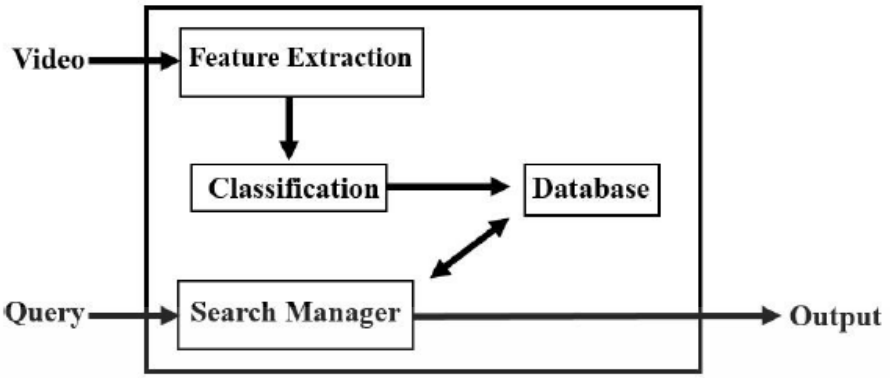

Figure 2.1: System overview

Type Input Features	Color Input Features
Bounding box • X coordinate from top-left corner • Y coordinate from top-left corner • Width • Height Ratio of vehicle over background	Color in HSV space • Hue • Saturation • Value

Figure 2.2: Input features in vehicle type and vehicle color classification task

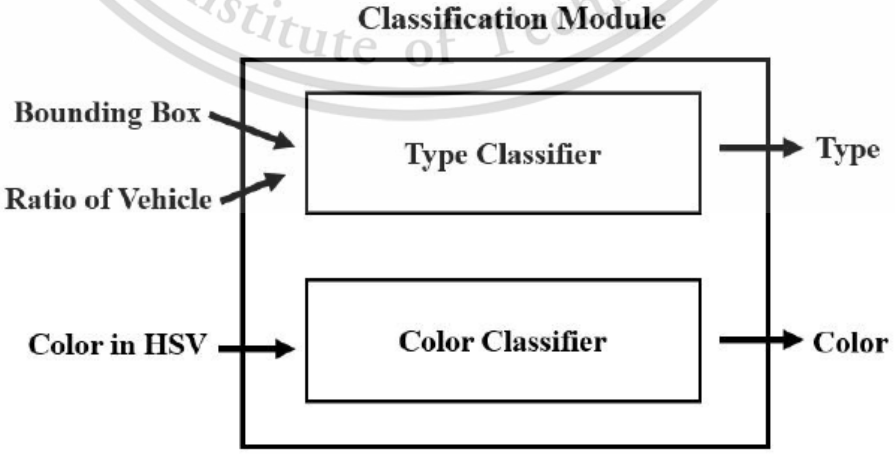

Figure 2.3: Classification module overview

Type Output Classes	Color Output Classes
Small	Black
Medium	White
Large	Red
Unknown	Blue
	Yellow
	Green
	Unknown

Figure 2.4: Output classes in vehicle type and vehicle color classification task

Method	Accuracy(%)	
	Type	Color
Decision tree	79.38	67.98
Adaboost	78.94	**69.29**
Bagging	77.63	68.85
Random forest	**79.82**	68.42

Figure 2.5: Experiment result

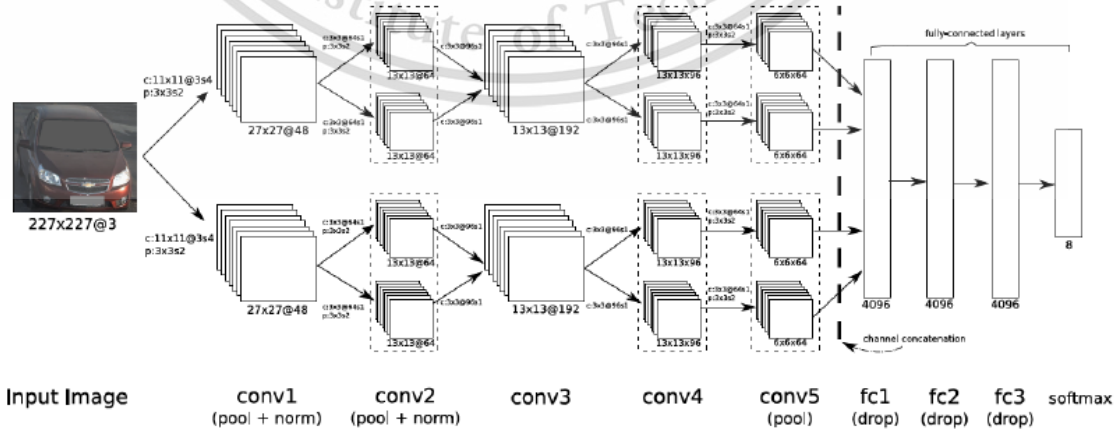

Figure 2.6: The CNN structure

	Color Space				Chen et al. [2]
Color Class	RGB	HSV	CIE Lab	CIE XYZ	
yellow	0.9794	0.9450	0.9656	**0.9828**	0.9553
white	**0.9666**	0.9624	0.9561	0.9649	0.9423
blue	0.9410	**0.9576**	0.9410	0.9484	0.9535
cyan	0.9645	0.9716	0.9645	0.9716	**0.9787**
red	**0.9897**	0.9866	**0.9897**	0.9886	0.9878
gray	0.8608	0.8503	**0.8668**	0.8647	0.8466
black	**0.9738**	0.9703	0.9703	0.9709	0.9730
green	**0.8257**	0.8215	0.8215	0.7676	0.7884
average	**0.9447**	0.9372	0.9414	0.9432	0.9282

Figure 2.7: Experiment results

They used RELU(Rectified Linear Unit) as an activation function for all layers and max pooling with size 3x3 and stride2 for the pooling method. The sixth and seventh layer are fully-connected layers which consist of Dropout regularization method to prevent the overfitting problem. The last layer is Softmax. In experiment, they used stochastic gradient descent with 115 examples per batch, momentum of 0.9 and weight decay of 0.0005. Moreover, Gaussian function is used as a weight initialization. There are four colors spaces using in the experiment. The result is shown in Figure 2.7. From the table, the RGB color space achieved the best performance and even better than the previous model which is proposed by Chen 2% in term of accuracy. However, green and gray color achieve less than 90%. There is another table which is a probability table of color classification.

2.3 Image-based vehicle analysis using deep neural network: A systematic study

Zhou et al. [4] proposed the Deep Neural Networks(DNNs) approaches for vehicle detection and classification. In their work the detection and classification based on rear view images only. In content, there are three main topics which are DNN approach for vehicle detection, DNN approach for vehicle classification and classify methods on poor lighting conditions.

Firstly, they applied the YOLO [5] model which to their own dataset on. There are some errors in vehicle detection such as detect vehicles which are about to out of the region of interest or detect vehicle overlapped with other vehicles. However, they fixed this outliers by using Re-

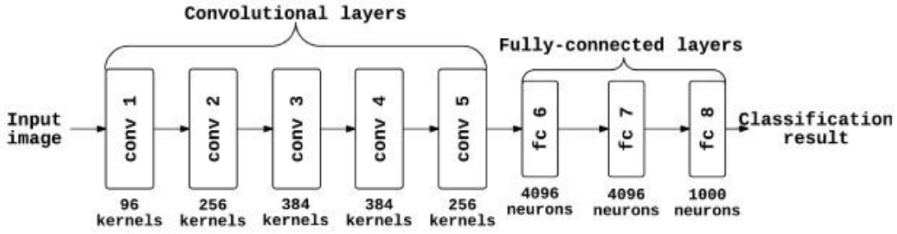

Figure 2.8: The image-based vehicle analysis structure

moving Outlier method. The second topic is DNN classification, there are two classes which are passenger and other. they applied the Alexnet [6] model as in the figure 2.8

The input image size is 256x256. They presented two approaches which are feature extraction and fine-tuning. The feature extraction approach, use Alexnet as a structure and extract layer fc6 or fc7 as a feature vector. When they obtained the vector, they apply it with SVM to be a classifier. In fine-tuning approach, also use Alexnet as a structure and change the size of layer fc8 from 1000 to 2 which match their dataset with two classes. Lastly, they do experiments in classification on poor lighting conditions by using two methods. The first method is scene transformation method and the another is Late fusion.

In conclusion, they proposed DNNs approaches for vehicle detection and classification which are improved from YOLO and Alexnet models. The performance of the system which are tuned model is better than the original for their work. In addition, Late fusion is better than scene transformation method in dark vehicle image classification.

2.4 Vehicle Color Recognition in The Surveillance with Deep Convolutional Neural Network

Su et al.[7] proposed new CNN structure named Colornet which achieved the highest accuracy in their vehicle color classification experiment of 95.74%. The structure outperformed Alexnet and GoogleNet [8]

They proposed a classifier named Colornet which is an eight layers network and sets Network In Network (NIN) [9] as an additional means. NIN is actually a network structure which can enhance the model discriminability. The traditional convolutional layers use linear filters followed by a nonlinear activiation fuction to scan the input. Although NIN uses more complex structures (micro neural network) instead to abstract the data. With enhanced local modeling via the micro network, we are able to utilize global average pooling over feature maps in the classification layer, which is easier to interpret and less prone to overfitting than traditional fully connected lay-

Method	Black	Blue	Gray	Green	Red	Cyan	White	Yellow	Average
C1	-	-	-	-	-	-	-	-	0.7880
C3	0.9495	0.9810	0.9535	0.9455	0.9615	0.9394	0.9286	0.9583	0.9522
C9	0.9220	0.9857	0.9651	0.9636	0.9846	0.9091	0.9847	0.9444	0.9574
C9, add*	0.9094	0.9227	0.3874	0.8465	0.8864	0.7801	0.9618	0.9829	0.8347
C9,add*-ft	0.9355	0.9761	0.8490	0.8797	0.9938	0.9858	0.9456	0.9622	0.9410

Note: C1 is Alexnet, C3 is GoogleNet and C9 is Colornet.

Figure 2.9: The color recognition performance comparison among CNN structures

ers. The authors built their own dataset. The images are collected from the HD bayonets, cropped from the surveillance videos and resized to 256×256. The dataset is consist of 15,016 vehicle images with different types, such as car, bus, SUV and truck. They set eight main colors for these images. The eight colors are black, blue, gray, green, red, cyan, white and yellow. In their experiment, the authors compared their proposed method named Colornet with the other well-known CNN stuctures, i.e. Alexnet and GoogleNet. The proposed method outperformed Alexnet and GoogleNet by approximately 17% and 0.5% respectively. The proposed method has far less layers than GoogleNet although with almost the same performance. The proposed method could save lots of computing resource and time because of the smaller size of the model. The result is shown in the figure2.9

CHAPTER 3

BACKGROUND KNOWLEDGE

This chapter explains about some background knowledge of deep learning and Convolutional Neural Network. The detail is presented in the following sections.

3.1 Deep Learning

In recent decades, there is another method call Deep Learning [10] which can be used in classification task. Deep learning is a neural network with more than two hidden layers. Following are some of the facets in evolution of neural network:

- More neurons than previous networks
- More complex ways of connecting layers/neurons in neural networks
- Explosion in the amount of computing power available to train
- Automatic feature extraction

There are many types of deep learning, e.g. unsupervised pretrained networks, convolution neural networks (CNN), recurrent neural networks, recursive neural network, etc. Convolutional neural networks are the most popular neural network in deep learning. The main characteristic of these networks is convolution, which is designed to learn higher features in the data. The networks are well suited to object recognition with images and consistently top classifier in image classification competitions. Krizhevsky et al. [6] is the most popular CNN which won the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2012. The efficacy of CNNs in image recognition is one of the main reasons why the world recognizes the power of deep learning.

3.2 Convolutional Neural Network

Convolutional Neural Network is a kind of feed forward artificial neural network, it is quite a similar to standard neural network. The neurons in the network have learnable weights and biases. Every neuron receives inputs and performs some operations. There are three major layers in CNN, i.e. convolutional layer, pooling layer, and fully connected layer. convolutional layer will calculate the output of neurons that are connected to local regions within the input, each computes a dot product between the weights and biases. Pooling layer is used in order to reduce the feature maps size. It means that the parameters will be reduced too, the computation time is than faster. In general, max pooling is used in CNN. In fully connected layer, each neuron in this layer is

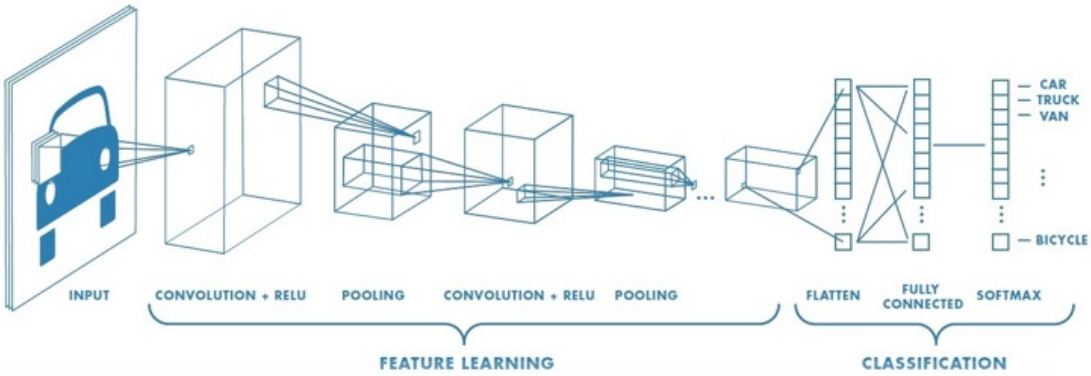

Figure 3.1: Typical CNN structure

connected to previous layer neurons. The layers are fully connected as in the same manner as in a common neural network. Figure 3.1 shows the general CNN structure. In feature learning section, this part is convolution, one main characteristics of CNN. The convolution makes CNN create its own features which means CNN do not need feature extraction in order to learn the information from an input.

In convolution, first the CNN structure is feed by input image. Then the convolutional layer start doing the convolution. Figure 3.2 shows how does convolution work. There is a color input image (red rectangular box) which its size is $32 \times 32 \times 3$. In the figure, the convolutional layer's filter size is $5 \times 5 \times 3$. The filter moves to the right with a certain stride value (moves how many pixels in one step) until it parses the complete width. Moving on, it hops down to the beginning (left) of the image with the same stride value and repeats the process until the entire image is traversed. In addition, blue circle in the mentioned figure represents one pixel on activation map (blue rectangular box). The purpose of the convolution is to get new representation of the input which is an activation map. Suppose the convolutional layer has 6 filters, the convolution's output are 6 activation maps as shown in the figure 3.3. Figure 3.4 shows the sample output of the convolution in real image. The objective of the convolution is to extract the high-level features such as edges, from the input image. CNN need not be limited to only one convolutional layer. Conventionally, the first convolutional layer is responsible for capturing the low-level features such as edges, color, gradient orientation, etc. with added layers, the architecture adapts to the high-level features as well, giving us a network which has the well understanding of images in the dataset, similar to how we would.

Similar to the convolutional layer, the pooling layer is responsible for reducing the spatial size of the convolved feature. This is to decrease the computational power required to process the

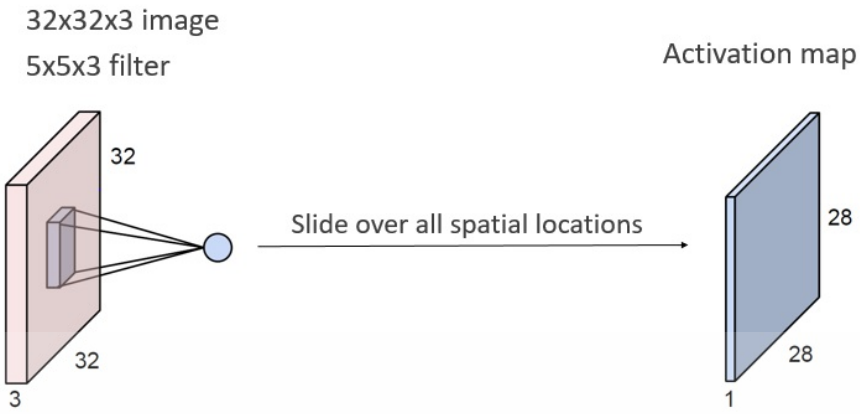

Figure 3.2: Convolution in convolutional layer

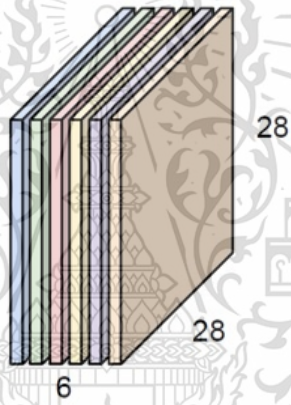

Figure 3.3: Activation maps

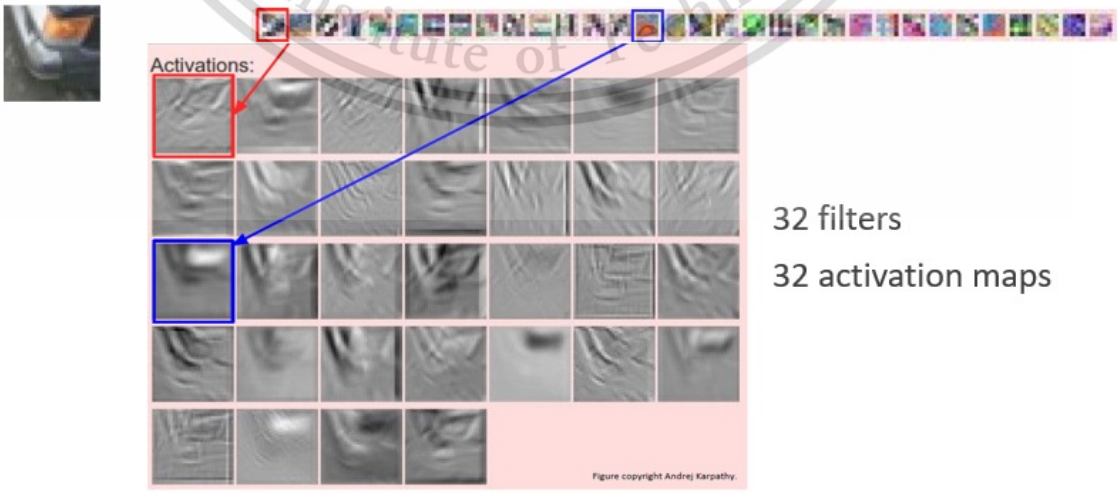

Figure 3.4: Sample outputs of convolution

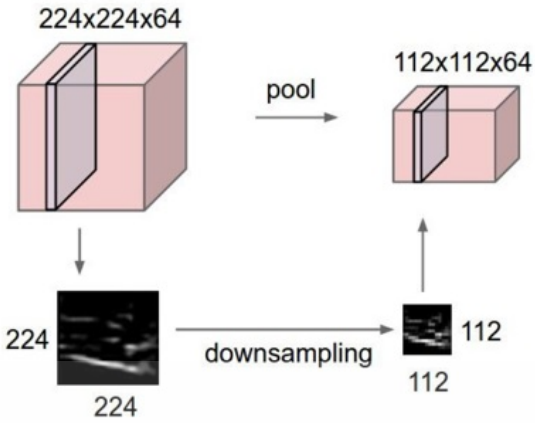

Figure 3.5: Pooling operation

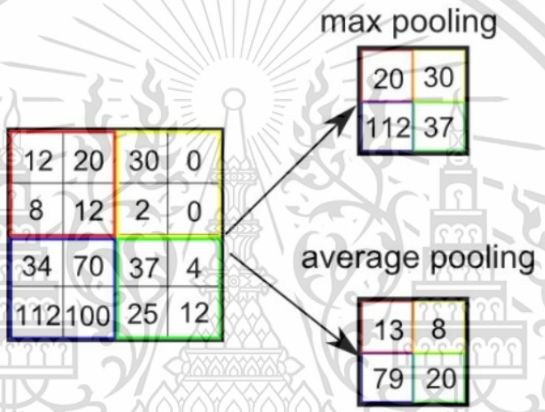

Figure 3.6: Max pooling and average pooling

data through dimensionality reduction. Furthermore, it is useful for extracting dominant features which are rotational and positional invariant, thus maintaining the process of effectively training of the model. Figure 3.5 shows the pooling operation. There are two types of pooling, i.e. max pooling and average pooling. Max pooling returns the maximum value from the portion of the image covered by the kernel. On the other hand, average Pooling returns the average of all the values from the portion of the image covered by the kernel. Figure 3.6 shows examples of the max pooling operation and average pooling operation.

CHAPTER 4

METHODOLOGY

This chapter introduces the methods use and conduct in this study. The methods which are used both are Convolutional Neural Network. There are two main sections, i.e. two-leveled CNN and standard CNN structures. The detail of each section will be explained.

4.1 Two-leveled Convolutional Neural Network

Two-leveled Convolutional Neural Network. The proposed CNN architectures contain the following characteristics which are shown in Table III. There are two convolution layers, i.e. 1st and 2nd layer. In the structure, pooling layers are set at the positions after each convolution layer which is already applied with activation function. The activation function is used to operate after the process of convolution and fully connected, although the last layer is not applied the function. The 3rd, 4th and 5th layer are fully connected layers. Dropout is a common method which can be used to avoid overfitting. Output or predictor is the final layer. The number of neurons in this layer is equal to the number of possible classes. Moreover, the outputs are predicted classes with probability score in range 0 to 1.

Figure 4.1. shows the structure of the CNN. Firstly, the original vehicle image is resized into 3232 pixels by using resize function in Tensorflow. The resized image is fed to the 1st convolutional layer with 32 filters of size 443 and stride of one pixel. The output of the first convolution layer is modeled by ReLU. Max pooling reduces the size of the feature map outputs with kernel size of 22 and the stride of two pixels. Then the output is passed to the 2nd convolutional layer and the same operations are repeated. After that, the output of 2nd convolution layer is converted into vector form and then fed into fully connected layer. In the fully connected layers, there are dot product operation and ReLU operation, respectively. The dropout is added at the position after the 4th layer in order to prevent the overfitting in training networks. The final layer is a predictor which is softmax. In addition, a number of neurons are equal to a number of possible classes. In this work, the last layer of vehicle type classification contains four neurons and there are seven neurons at last layer of vehicle color classification.

4.2 Standard CNN Stuctures

There are several kinds of CNN structures. An image dataset and a complex problem are required in order to test the performance of those structures. There is a famous dataset named

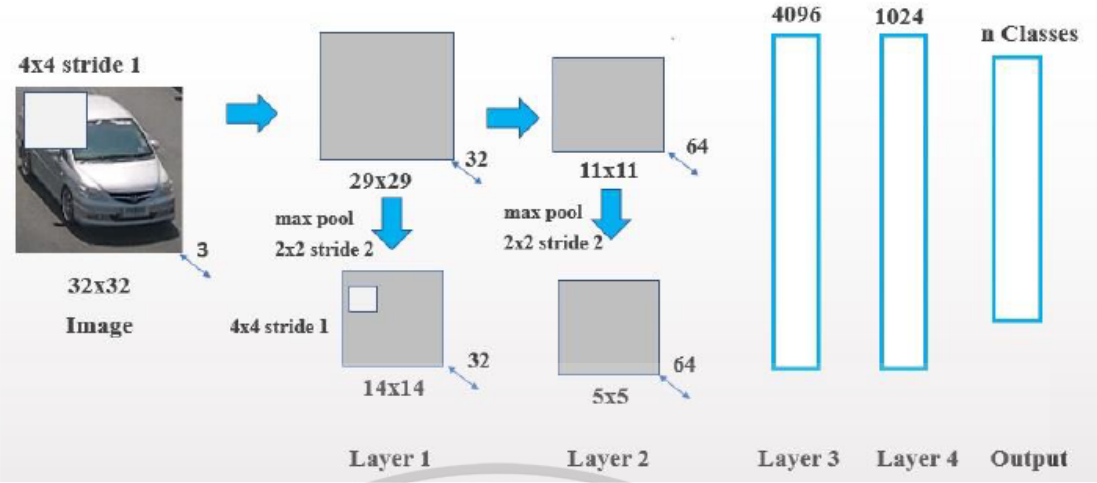

Figure 4.1: Two-leveled CNN structure

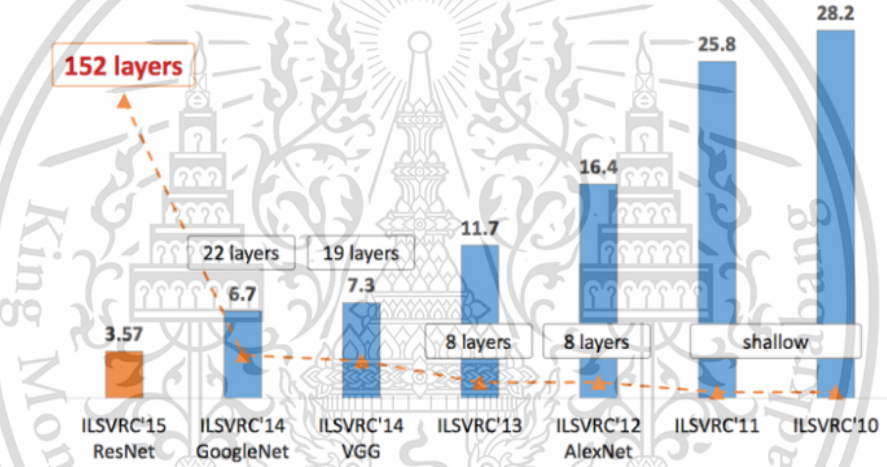

Figure 4.2: ILSVRC challenge results (top-5 error(%)) from 2010 to 2015

ImageNet, consists of over 15 millions labeled high-resolution images with around 22,000 categories. ILSVRC(ImageNet Large Scale Visual Recognition Competition) uses a subset of ImageNet aroud 1,000 images in each of 1,000 categories. In all, there are roughly 1.2 million training images, 50,000 validation images and 100,000 testing images. Figure 4.2 shows the competition results from 2010 to 2015. In this research, four famous CNN model from the ILSVRC competiton are used i.e. Alexnet, VGG, Inception (GoogleNet) and ResNet.

4.2.1 Alexnet

Alexnet is one of the most famous among Convolutional Neural Network models. Krizhevsky et al [6] 's work or Alexnet won the ILSVRC (ImageNet Large Scale Visual Recognition Challange) 2012 award. The paper used a CNN to get a Top-5 error rate (rate of not finding the true label of a given image among its top 5 predictions) of 15.3%. The next best result trailed

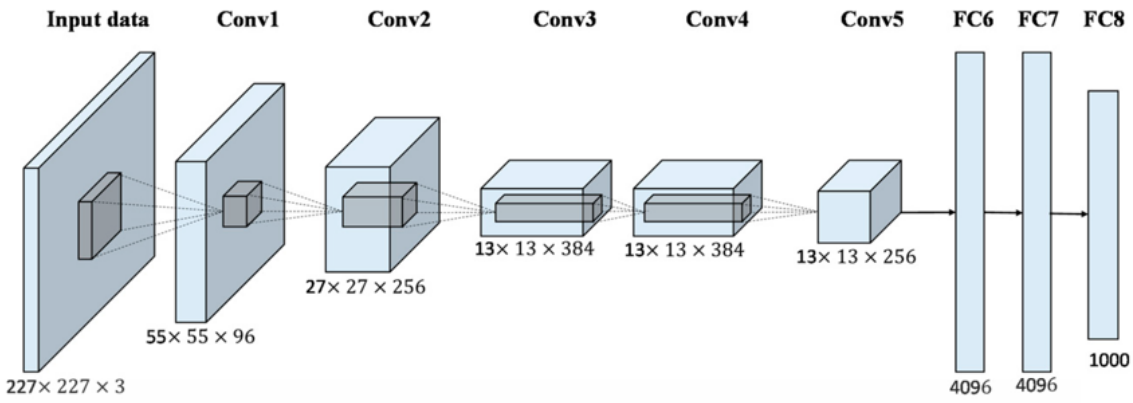

Figure 4.3: Alexnet structure

far behind 26.2%. Deep Learning became more popular since that day. In next few years, mutiple teams would build CNN architectures that beat human level accuarcy. The architecture used in the 2012 paper is called Alexnet. The input to AlexNet is an RGB image of size 256256. This means all images in the training set and all test images need to be of size 256256. If the input image is not 256256, it needs to be resized to 256256 before using it for training the network. The input images are randomly cropped to be size of 227x227 later. AlexNet consists of 5 Convolutional Layers and 3 Fully Connected Layers. Figure 4.3 shows the original structure of Alexnet. In addition, the followings are the major hyperparameters of the structure:

- Max pooling operation is used in pooling layer
- ReLU (Rectified Linear Unit) is an activation function that used in the structure
- The optimizer is Gradient descent optimizer
- Softmax is used as the predictor
- Dropout rate is 0.5
- Three fully connected layers consists of 4096, 4096 and 1,000 neurons respectively.

In this research, the optimizer is changed from Gradient descent optimizer to Adam optimizer. In these days, Adam optimizer is better than Gradient descent optimizer. Adam optimizier combine the best properities of the AdaGrad and RMSProp algorimithms to provide an optimization algorithm that can handle sparse gradients on noisy problems. In addition, the numbers of neurons in the last fully connected layers is changed from 1,000 to 4 or 7 for vehicle type classification and vehicle color classification respectively.

4.2.2 VGG

This architecture is from VGG(Visual Geometry Group), department of engineering science, university of Oxford. The reference of the mentioned structure is [11]. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3×3 kernel-sized filters one after another. With a given receptive field(the effective area size of input image on which output depends), multiple stacked smaller size kernel is better than the one with a larger size kernel because multiple non-linear layers increases the depth of the network which enables it to learn more complex features, and that too at a lower cost.

For example, three 3×3 filters on top of each other with stride 1 ha a receptive size of 7, but the number of parameters involved is $3 \times (9C^2)$ in comparison to $49C^2$ parameters of kernels with a size of 7. Here, it is assumed that the number of input and output channel of layers is C.Also, 3×3 kernels help in retaining finer level properties of the image. The network architecture is given in the table figure 4.4. In VGG-D column, there are blocks with same filter size applied multiple times to extract more complex and representative features. This concept of blocks/modules became a common theme in the networks after VGG. The VGG convolutional layers are followed by 3 fully connected layers. The width of the network starts at a small value of 64 and increases by a factor of 2 after every sub-sampling/pooling layer. It achieved the top-5 accuracy of 92.3 % on ImageNet.

In this research, The VGG-E column from the mention table figure, called VGG-19, is used as one of CNN structures in the experiment. VGG-19 is the current latest version. Furthermore, the numbers of neurons in the last fully connected layers is changed from 1,000 to 4 or 7 for vehicle type classification and vehicle color classification respectively.

4.2.3 Inception/GoogleNet

While VGG achieves a phenomenal accuracy on ImageNet dataset, its deployment on even the most modest sized GPUs is a problem because of huge computational requirements, both in terms of memory and time. It becomes inefficient due to large width of convolutional layers. For instance, a convolutional layer with 3×3 kernel size which takes 512 channels as input and outputs 512 channels, the order of calculations is $9 \times 512 \times 512$.

In a convolutional operation at one location, every output channel (512 in the example above), is connected to every input channel, and so we call it a dense connection architecture. The GoogleNet builds on the idea that most of the activations in a deep network are either unnecessary

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 4.4: VGG structure

(value of zero) or redundant because of correlations between them. Therefore the most efficient architecture of a deep network will have a sparse connection between the activations, which implies that all 512 output channels will not have a connection with all the 512 input channels. There are techniques to prune out such connections which would result in a sparse weight/connection. But kernels for sparse matrix multiplication are not optimized in BLAS or CuBlas(CUDA for GPU) packages which render them to be even slower than their dense counterparts.

So GoogleNet devised a module called inception module that approximates a sparse CNN with a normal dense construction (shown in the figure 4.5). Since only a small number of neurons are effective as mentioned earlier, the width/number of the convolutional filters of a particular kernel size is kept small. Also, it uses convolutions of different sizes to capture details at varied scales(5×5, 3×3, 1×1).

Another salient point about the module is that it has a so-called bottleneck layer(1X1 convolutions in the figure 4.5). It helps in the massive reduction of the computation requirement as explained below.

"Let us take the first inception module of GoogleNet as an example which has 192 channels as input. It has just 128 filters of 3×3 kernel size and 32 filters of 5×5 size. The order of computation for 5×5 filters is $25 \times 32 \times 192$ which can blow up as we go deeper into the network when the width of the network and the number of 5×5 filter further increases. In order to avoid this, the inception module uses 1×1 convolutions before applying larger sized kernels to reduce the dimension of the input channels, before feeding into those convolutions. So in the first inception module, the input to the module is first fed into 1×1 convolutions with just 16 filters before it is fed into 5×5 convolutions. This reduces the computations to $16 \times 192 + 25 \times 32 \times 16$. All these changes allow the network to have a large width and depth."

Another change that GoogleNet made, was to replace the fully connected layers at the end with a simple global average pooling which averages out the channel values across the 2D feature map, after the last convolutional layer. This drastically reduces the total number of parameters. This can be understood from AlexNet, where FC layers contain approx. 90% of parameters. Use of a large network width and depth allows GoogleNet to remove the FC layers without affecting the accuracy. It achieves 93.3% top-5 accuracy on ImageNet and is much faster than VGG.

Figure 4.6 shows the overall structure of GoogleNet or Inception architecture. In this research, InceptionV3 is used in the experiment. The lastest version of Inception is InceptionV3. In addition, the numbers of neurons in the last fully connected layers is changed from 1,000 to 4 or

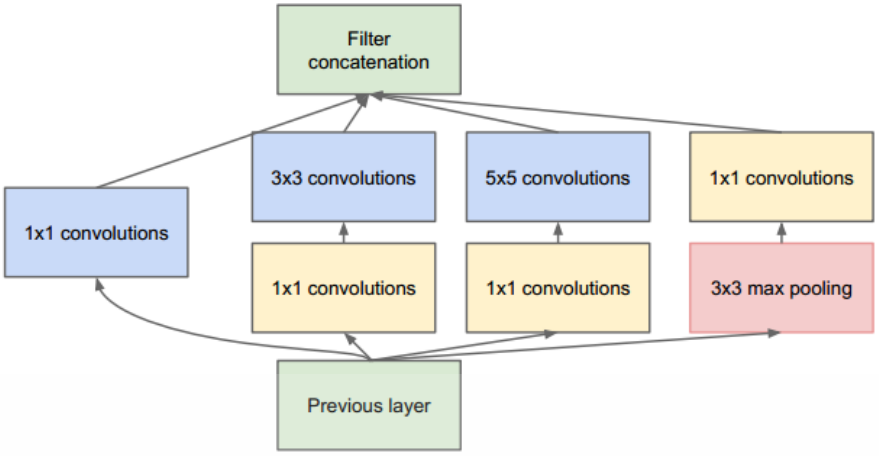

Figure 4.5: Inception module

7 for vehicle type classification and vehicle color classification respectively.

4.2.4 ResNet

In this section, the significant information of ResNet [12] will be explained. As per what we have seen so far, increasing the depth should increase the accuracy of the network, as long as over-fitting is taken care of. But the problem with increased depth is that the signal required to change the weights, which arises from the end of the network by comparing ground-truth and prediction becomes very small at the earlier layers, because of increased depth. It essentially means that earlier layers are almost negligible learned. This is called vanishing gradient. The second problem with training the deeper networks is, performing the optimization on huge parameter space and therefore naively adding the layers leading to higher training error. Residual networks allow training of such deep networks by constructing the network through modules called residual models as shown in the figure 4.7

Imagine a network, suppose A which produces x amount of training error. Construct a network B by adding few layers on top of A and put parameter values in those layers in such a way that they do nothing to the outputs from A. Lets call the additional layer as C. This would mean the same x amount of training error for the new network. So while training network B, the training error should not be above the training error of A. And since it does happen, the only reason is that learning the identity mapping(doing nothing to inputs and just copying as it is) with the added layers-C is not a trivial problem, which the solver does not achieve. In order to solve this, the module which is shown in figure 4.7 creates a direct path between the input and output to the module implying an identity mapping and the added layer-C just need to learn the features on top of already available input. Since C is learning only the residual, the whole module is called

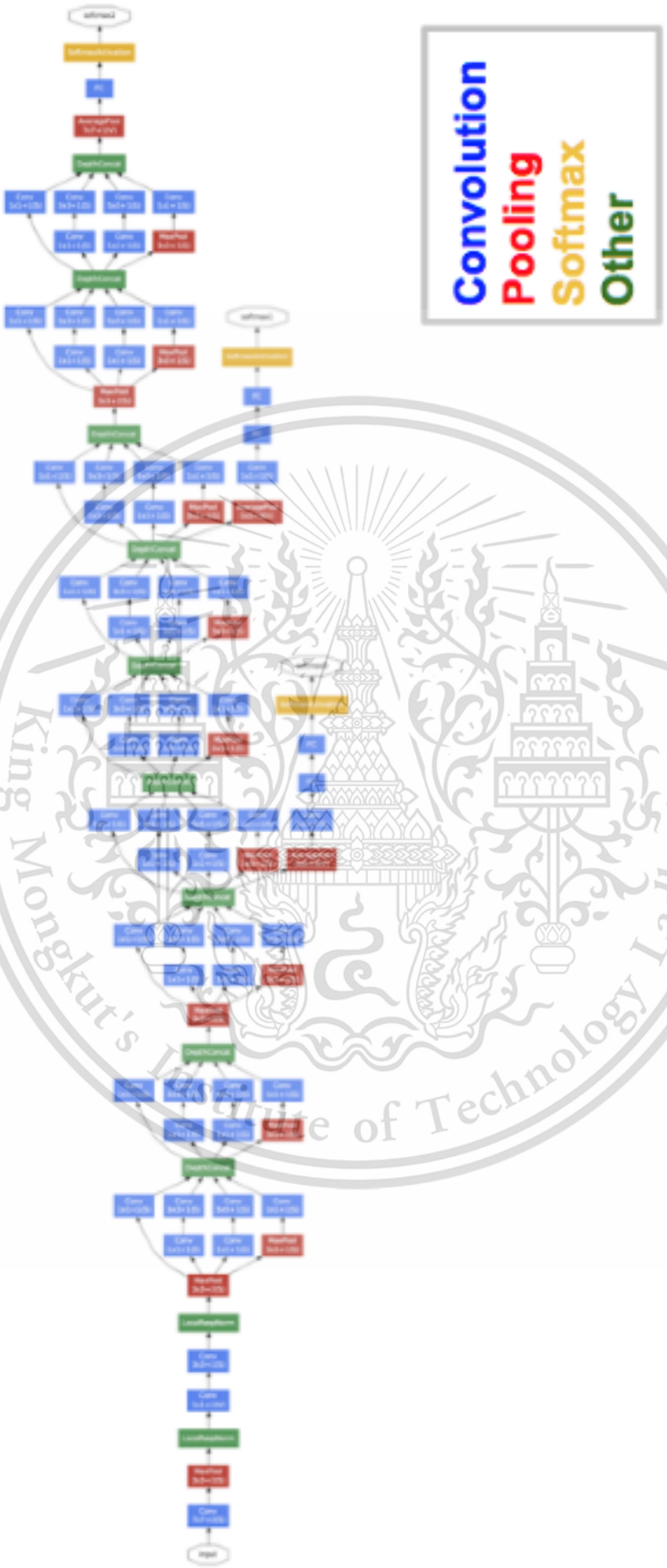

Figure 4.6: GoogleNet structure

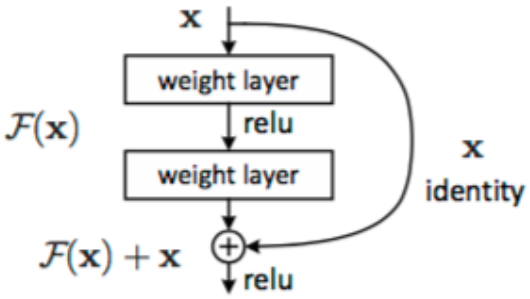

Figure 4.7: Residual learning: a building block

residual module or residual block.

In addition, it is similar to GoogleNet, it uses a global average pooling followed by the classification layer. Through the changes mentioned, ResNets were learned with network depth of as large as 152. It achieves better accuracy than VGGNet and GoogleNet while being computationally more efficient than VGGNet. ResNet-152 achieves 95.51% top-5 accuracies. The architecture is similar to the VGGNet consisting mostly of 3×3 filters. From the VGGNet, shortcut connection as described above is inserted to form a residual network. This can be seen in the figure which shows a small snippet of earlier layer synthesis from VGG-19.

Figure 4.8 shows the comparison between residual network and plain network (for example, Alexnet). Each colored block of layers represent a series of convolutions of the same dimension. The feature mapping is periodically downsampled by strided convolution accompanied by an increase in channel depth to preserve the time complexity per layer. Dotted lines denote residual connections in which we project the input via a 1×1 convolution to match the dimensions of the new block. The mentioned figure 4.8 visualizes the ResNet 34 architecture. For the ResNet 50 model, we simply replace each two layer residual block with a three layer bottleneck block which uses 1×1 convolutions to reduce and subsequently restore the channel depth, allowing for a reduced computational load when calculating the 3×3 convolution. The difference between ResNet34 residual block and ResNet50 residual block is shown in figure 4.9. In this research, ResNet50 is conducted in the experiment. In addition, the numbers of neurons in the last fully connected layers is changed from 1,000 to 4 or 7 for vehicle type classification and vehicle color classification respectively.

4.3 Pretrained Weight

In order to improve more performance, the CNN structure has to be more deeper and more suitable. In short period, pretrained model is one of the good choice to make the large CNN

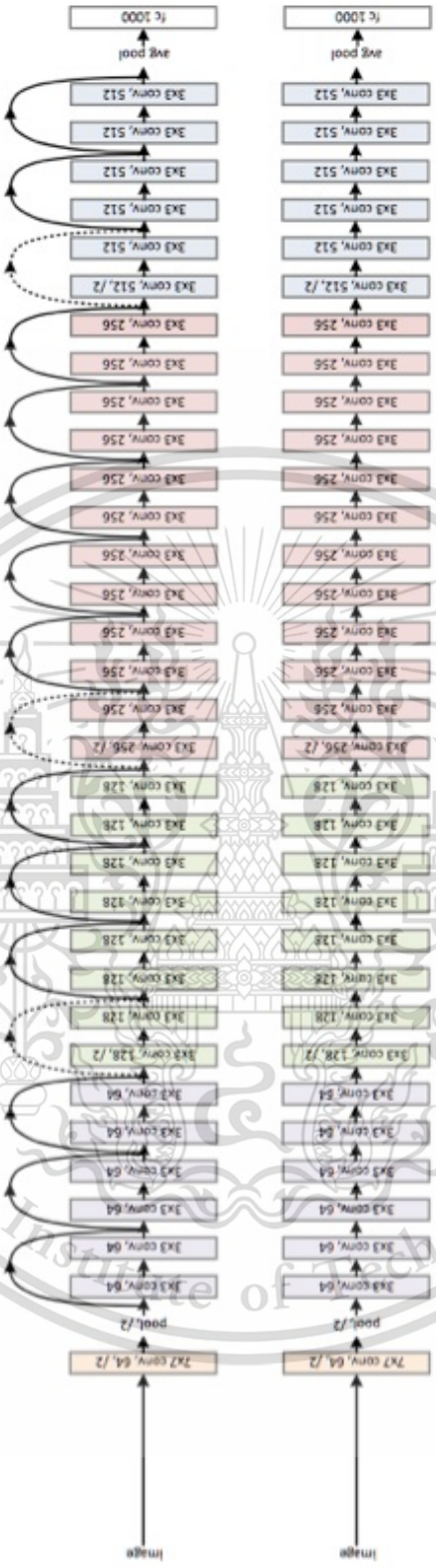

Figure 4.8: The difference between residual network and plain network

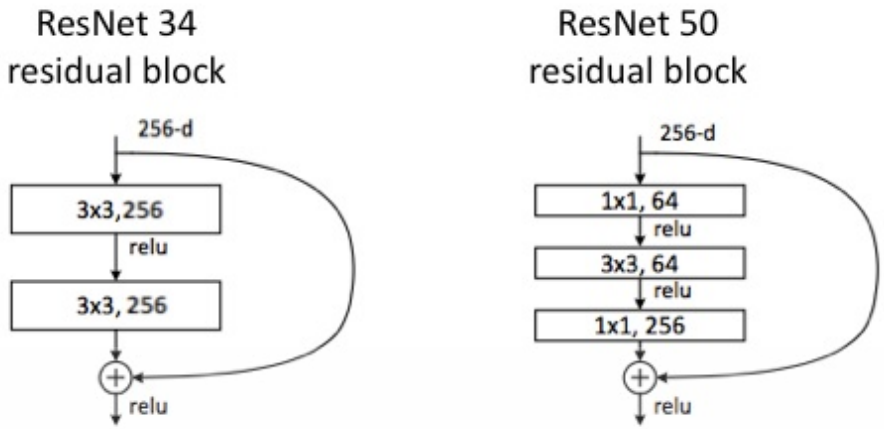

Figure 4.9: The difference between ResNet34 and ResNet50 residual block

structure train faster than usual. In this research, the weights of the model which have been trained by using ImageNet dataset are used as initial weights . The dataset is very large which contains more than 14 millions images and more categories. In addition, each category consists of several hundred images.

4.4 Data Augmentation

Data augmentation is the creation of altered copies of each instance within a training dataset. When the image data is feed into a neural network, there are some features of the images that the neural network does not need to in incorporate in its set of weights. In the case of image classification, these feature or noise are the pixels which form the background in the picture. The data augmentation makes the network could differentiate signal from noise. It creates multiple alterations of each image, where the signal or the object in the picture is kept invariant, whilst the noise or the background is distorted. These distortions include cropping, scaling and rotating the image, among others. Therefore, the network of neuron observes the invariant in the images and encodes this information or signal in the set of weights which summarize the training data. Figure 4.10 shows sample vehicle images with augmentation which used in this research. The original image is on the left hand while six images on the right are augmented images.

The purpose of using data augmentation is improving a dataset. Some datasets are not perfect, are lack of large amounts of data or have imbalance numbers of data among categories. The data augmentation can be utilized to solve those problems. It can help to increase the amount of relevant data in the dataset.

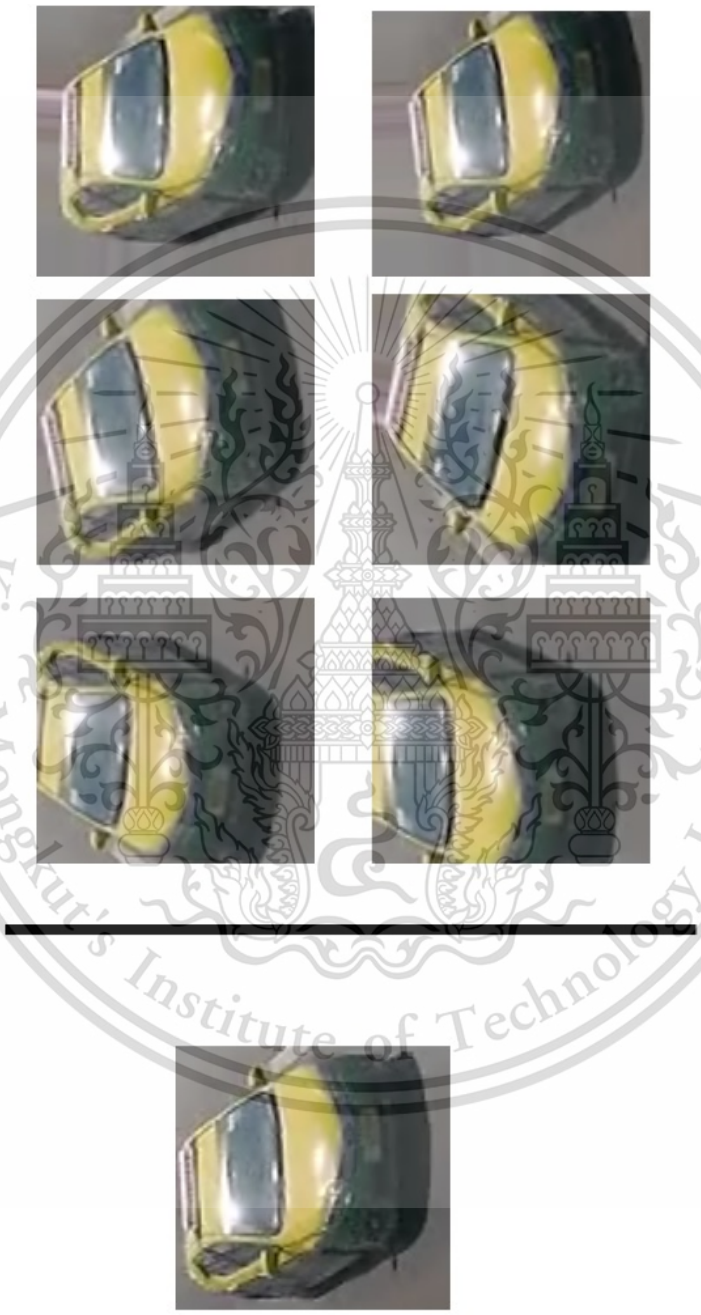

Figure 4.10: Some sample vehicle images with augmentation

CHAPTER 5

EXPERIMENTATION AND RESULTS

In this section, there are explanations and lists of environment and tools which used in this research. In addition, processes of development each CNN models are further explained in detail. The experiments are conducted in two environments, i.e. a specific computer and Googles GPU. The results achieved in each experiment are summarized and discussed.

5.1 Experimental Environment

There are two environments which mentioned above. The experiments which are conducted on a specific computer consist of Two-level CNN and Alexnet. The information of the computer is shown in Table 5.1. These two CNN models are implemented by using Tensorflow. The remaining CNN models which are used in this research, i.e. VGG-19, InceptionV3, and ResNet50 is constructed on Google Colaboratory. Google Colaboratory provides GPU for free 5 hours. The detail of the GPU is mentioned in Table 5.2. Any developers could develop deep learning applications with Google Colaboratory using Keras, Tensorflow and Pytorch. GPU has chosen because the three CNN models have very large structure, this environment could save more time doing experiments with the huge models. The large CNN models in this research are implemented by using Keras. Furthurmore, Saripan [1] vehicle image dataset is used in this research. Figure 5.1 shows the sample vehicle image from the dataset.

5.2 Experiment 1: Vehicle Type Classification

In this type classification, the experiment is divided into two main parts, i.e. CNN models with pretrained weights and CNN models with no pretrained weights. Both standard CNN models

Processor	Intel (R) Core (TM) i5-4690 CPU Processor, 3.50 GHz clock frequency, 4 cores 8 threads
Physical Memory Capacity	16.0 GB
Physical Memory Type	DDR3
Physical Memory Speed Bus	799.5 MHz
Operating System	Microsoft Windows 10

Table 5.1: Specifications of the experimental computer

GPU	Tesla K80, 562 MHz GPU clock frequency, 4,992 cores
Physical Memory Capacity	12.0 GB
Physical Memory Type	GDDR5
Bandwidth	240.6 GB/s

Table 5.2: Specifications of Google Colaboratory's GPU

Figure 5.1: Sample vehicle images from Saripan [1] dataset

are run 4 times, the result of each model are average accuracy and standard deviation.

5.2.1 Objective

This experiment is conducted to measure and compare accuracies between different convolutional neural networks for different architectures for classifying a vehicle type. The approaches experimented are two-leveled CNN, Alexnet, VGG-19, InceptionV3, and ResNet50.

5.2.2 Experiment Setup 1.1 : Non-Pretrained Weights

The experiment is set up as follows:

The dataset from [1] is used in this experiment. There are 914 vehicle images. The dataset consists of 4 labeled classes, i.e. small, medium, large, and unknown. The vehicle images are resized and fed into the CNN models as follow:

- Two-leveled CNN : 32×32 pixels
- Alexnet : 224×224 pixels
- VGG-19 : 224×224 pixels
- InceptionV3 : 299×299 pixels
- ResNet50 : 224×224 pixels

The lists of inputs and labeled classes of this experiment can also be seen in Table 5.3. Each vehicle image is color images which refers to the input size of $N \times N \times 3$, $N \times N$ is size of resized image which mentioned above. In addition, 75% of the dataset is divide to be used as training set and the remaining 25% is used in testing. The number of epochs of each model is set to be 50 except Two-leveled CNN 's. 500 is the number of epochs of Two-leveled CNN. Furthermore, this experiment is also designed to compared the difference between original dataset and original dataset with data augmentation. In this experiment the augmented dataset consists of 300 vehicles images of each vehicle type class. Total number of vehicle images of the augmented dataset is 1,200.

5.2.3 Experiment Setup 1.1 : Results

The resulted performances are measured in testing accuracy. Testing accuracy is measured in percentage. It is calculated by dividing a number of correctly classified vehicle type by a number of all vehicle times one hundred. Thus, the possible range of testing accuracy is from 0 to 100. The details of results are shown in Table 5.4, InceptionV3 model with augmented dataset

Input	Labeled Classes
Resized vehicle images	Small
	Medium
	Large
	Unknown

Table 5.3: List of inputs and labeled classes in vehicle type classification

Model	Original Dataset		Augmented Dataset	
	Testing Accuracy(%)	STD	Testing Accuracy(%)	STD
Two-leveled CNN	81.56	2.597	82.89	2.453
Alexnet	81.47	1.358	79.72	1.206
VGG-19	90.46	1.576	90.90	1.309
InceptionV3	91.56	0.554	**92.33**	2.042
ResNet50	91.23	1.791	91.89	0.912

Table 5.4: Comparison table of testing accuracy and standard deviation of five CNN models with no pretrained weights for vehicle type classifying task

achieved the best accuracy with standard deviation of 2.042. From the Table 5.4, this table shows that augmented dataset improved the performance of very deep CNN (VGG-19, InceptionV3 and ResNet50). However, among those three deep CNNs, there is only InceptionV3 that had more unstable accuracy when using augmented dataset compare with using original dataset.

5.2.4 Experiment Setup 1.2 : Pretrained Weights

The experiment is set up as follows:

The dataset from [1] is used in this experiment. There are 914 vehicle images. The dataset consists of 4 labeled classes, i.e. small, medium, large, and unknown. The vehicle images are resized and fed into the CNN models as follow:

- Alexnet : 224×224 pixels
- VGG-19 : 224×224 pixels
- InceptionV3 : 299×299 pixels
- ResNet50 : 224×224 pixels

The lists of inputs and labeled classes of this experiment can also be seen in Table 5.3. Each

Model	Original Dataset		Augmented Dataset	
	Testing Accuracy(%)	STD	Testing Accuracy(%)	STD
Alexnet	84.32	0.976	87.61	0.66
VGG-19	**89.48**	0.618	87.50	0.564
InceptionV3	72.92	0.549	71.23	2.548
ResNet50	27.19	0	27.19	0

Table 5.5: Comparison table of testing accuracy and standard deviation of four CNN models with pretrained weights for vehicle type classifying task

vehicle image is color images which refers to the input size of $N \times N \times 3$, $N \times N$ is size of resized image which mentioned above. In addition, 75% of the dataset is divide to be used as training set and the remaining 25% is used in testing. The number of epochs of each model is set to be 500. Furthermore, this experiment is also designed to compared the difference between original dataset and original dataset with data augmentation. In this experiment the augmented dataset consists of 300 vehicles images of each vehicle type class. Total number of vehicle images of the augmented dataset is 1,200.

5.2.5 Experiment Setup 1.2 Results

The resulted performances are measured in testing accuracy. Testing accuracy is measured in percentage. It is calculated by dividing a number of correctly classified vehicle type by a number of all vehicle times one hundred. Thus, the possible range of testing accuracy is from 0 to 100. The details of results are shown in Table 5.5. VGG-19 model with original dataset achieved the best accuracy with standard deviation of 0.618. From the Table 5.5, this table shows that augmented dataset did not improve the performance of some models (VGG-19 and InceptionV3). In addition, all models which showed in the table are pretrained weight CNN models.

5.2.6 Summary

In this type classifying experiment, the best classifier is InceptionV3 model with augmented dataset, achieved an average accuracy of 92.33% with standard deviation of 2.042. The maximum accuracy of 94.3% from the 4-run experiment. Figure 5.2 shows training and validation accuracy graph of the maximum accuracy run. The confusion matrix of this run is shown in figure 5.3. Figure 5.4 also refers to the overall evaluation result of the 94.3% accurate run.

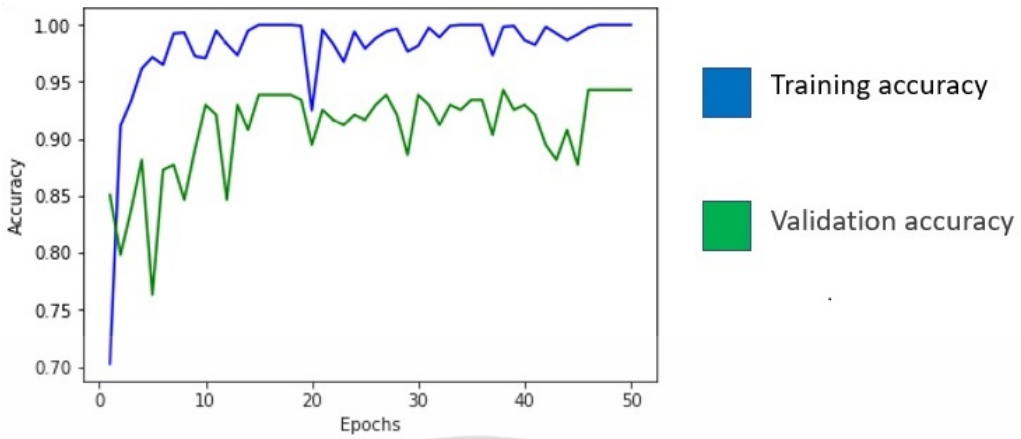

Figure 5.2: Training and validation accuracy graph of the best vehicle type classifier

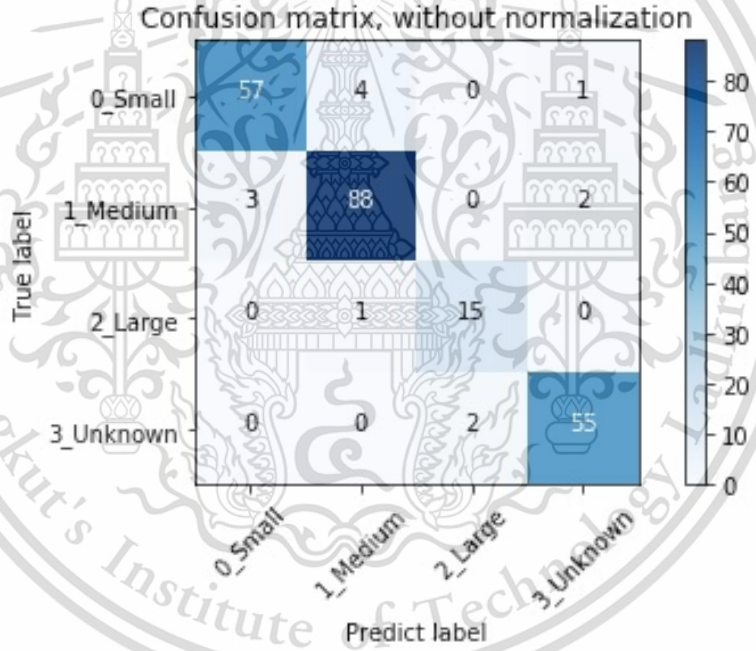

Figure 5.3: Confusion matrix of the best vehicle type classifier

	precision	recall	f1-score	support
0_Small	0.95	0.92	0.93	62
1_Medium	0.95	0.95	0.95	93
2_Large	0.88	0.94	0.91	16
3_Unknown	0.95	0.96	0.96	57

Figure 5.4: Overall evaluation result of the best vehicle type classifier

5.3 Experiment 2: Vehicle Color Classification

In this vehicle color classification, the experiment is divided into two main parts, i.e. CNN models with pretrained weights and CNN models with no pretrained weights. Both standard CNN models are run 4 times, the result of each model are average accuracy and standard deviation.

5.3.1 Objective

This experiment is conducted to measure and compare accuracies between different convolutional neural networks for different architectures for classifying a vehicle color. The approaches experimented are two-leveled CNN, Alexnet, VGG-19, InceptionV3, and ResNet50.

5.3.2 Experiment Setup 2.1 : Non-Pretrained Weights

The experiment is set up as follows:

The dataset from [1] is used in this experiment. There are 914 vehicle images. The dataset consists of 7 labeled classes, i.e. red, yellow, green, blue, black, white, and unknown. The vehicle images are resized and fed into the CNN models as follow:

- Two-leveled CNN : 32×32 pixels
- Alexnet : 224×224 pixels
- VGG-19 : 224×224 pixels
- InceptionV3 : 299×299 pixels
- ResNet50 : 224×224 pixels

The lists of inputs and labeled classes of this experiment can also be seen in Table 5.6. Each vehicle image is color images which refers to the input size of $N \times N \times 3$, $N \times N$ is size of resized image which mentioned above. In addition, 75% of the dataset is divide to be used as training set and the remaining 25% is used in testing. The number of epochs of each model is set to be 50 except Two-leveled CNN 's. 500 is the number of epochs of Two-leveled CNN. Furthermore, this experiment is also designed to compared the difference between original dataset and original dataset with data augmentation. In this experiment the augmented dataset consists of 270 vehicles images of each vehicle color class. Total number of vehicle images of the augmented dataset is 1,890.

Input	Labeled Classes
Resized vehicle images	Black
	White
	Red
	Blue
	Yellow
	Green
	Unknown

Table 5.6: List of inputs and labeled classes in vehicle color classification

Model	Original Dataset		Augmented Dataset	
	Testing Accuracy(%)	STD	Testing Accuracy(%)	STD
Two-leveled CNN	70.09	2.27	70.68	1.136
Alexnet	76.97	1.845	80.92	2.222
VGG-19	76.21	3.346	77.30	2.25
InceptionV3	82.46	2.453	83.55	2.833
ResNet50	81.91	3.995	**86.08**	1.493

Table 5.7: Comparison table of testing accuracy and standard deviation of five CNN models with no pretrained weights for vehicle color classifying task

5.3.3 Experiment Setup 2.1 : Results

The resulted performances are measured in testing accuracy. Testing accuracy is measured in percentage. It is calculated by dividing a number of correctly classified vehicle type by a number of all vehicle times one hundred. Thus, the possible range of testing accuracy is from 0 to 100. The details of results are shown in Table 5.7. ResNet50 model with augmented dataset achieves the best accuracy with standard deviation of 1.493. From the table 5.7, the table shows that the augmented dataset improve the performance of all models. From this experiment, we can conclude that augmented dataset is very helpful for the non-pretrained weight CNN models in vehicle color classification.

5.3.4 Experiment Setup 2.2 : Pretrained Weights

The experiment is set up as follows:

The dataset from [1] is used in this experiment. There are 914 vehicle images. The dataset consists of 7 labeled classes, i.e. red, yellow, green, blue, black, white, and unknown. The vehicle

images are resized and fed into the CNN models as follow:

- Alexnet : 224×224 pixels
- VGG-19 : 224×224 pixels
- InceptionV3 : 299×299 pixels
- ResNet50 : 224×224 pixels

The lists of inputs and labeled classes of this experiment can also be seen in Table 5.3. Each vehicle image is color images which refers to the input size of $N \times N \times 3$, $N \times N$ is size of resized image which mentioned above. In addition, 75% of the dataset is divide to be used as training set and the remaining 25% is used in testing. The number of epochs of each model is set to be 500. Furthermore, this experiment is also designed to compared the difference between original dataset and original dataset with data augmentation. In this experiment the augmented dataset consists of 270 vehicles images of each vehicle color class. Total number of vehicle images of the augmented dataset is 1,890.

5.3.5 Experiment Setup 2.2 Results

The resulted performances are measured in testing accuracy. Testing accuracy is measured in percentage. It is calculated by dividing a number of correctly classified vehicle type by a number of all vehicle times one hundred. Thus, the possible range of testing accuracy is from 0 to 100. The details of results are shown in Table 5.8. Alexnet model with augmented dataset achieves the best accuracy. From the Table 5.8, the table shows that the augmented dataset reduced the performance of VGG-19 and InceptionV3 models. From the experiment 1.2 and 2.2, we can conclude that augmented dataset can not help improving the performance of pretrained weight VGG-19 and InceptionV3 models.

5.3.6 Summary

In this color classifying experiment, the best classifier is ResNet50 model with augmented dataset, achieved an average accuracy of 86.08% with standard deviation of 1.493. The maximum accuracy of 88.16% from the 4-run experiment. Figure 5.5 shows training and validation accuracy graph of the maximum accuracy run. The confusion matrix of this run is shown in figure 5.6. Figure 5.7 also refers to the overall evaluation result of the 88.16% accurate run.

Model	Original Dataset		Augmented Dataset	
	Testing Accuracy(%)	STD	Testing Accuracy(%)	STD
Alexnet	75.77	0.972	**78.07**	1.136
VGG-19	77.63	0.508	77.30	0.903
InceptionV3	52.30	4.364	43.97	4.838
ResNet50	46.05	0	46.05	0

Table 5.8: Comparison table of testing accuracy and standard deviation of four CNN models with pretrained weights for vehicle color classifying task

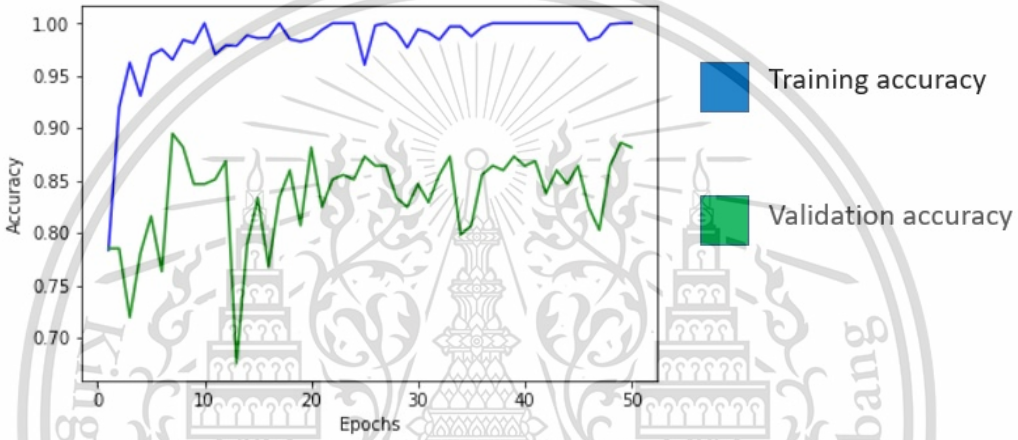

Figure 5.5: Training and validation accuracy graph of the best vehilce color classifier

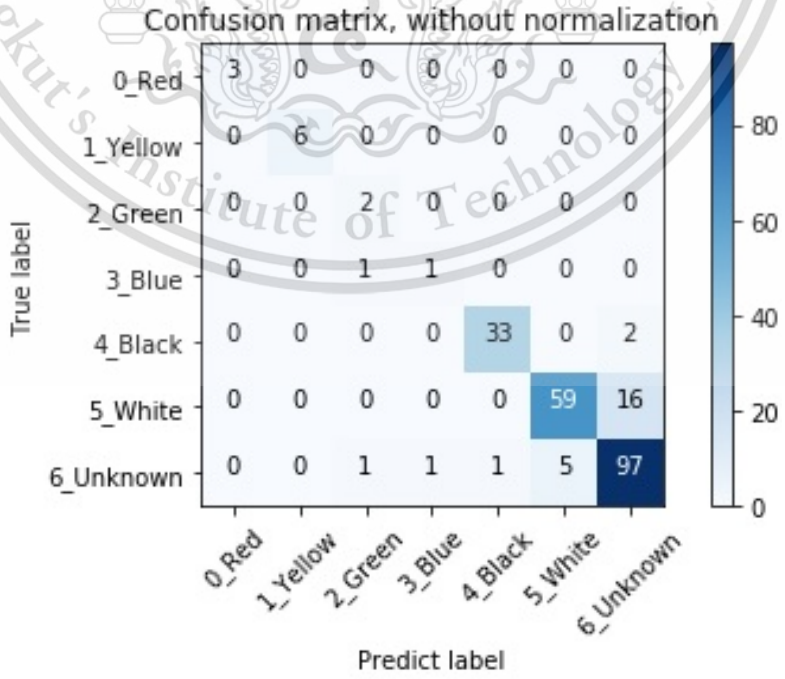

Figure 5.6: Confusion matrix of the best vehicle color classifier

	precision	recall	f1-score	support
0_Red	1.00	1.00	1.00	3
1_Yellow	1.00	1.00	1.00	6
2_Green	0.50	1.00	0.67	2
3_Blue	0.50	0.50	0.50	2
4_Black	0.97	0.94	0.96	35
5_White	0.92	0.79	0.85	75
6_Unknown	0.84	0.92	0.88	105

Figure 5.7: Overall evaluation result of the best vehicle color classifier

CHAPTER 6

CONCLUSION AND DISCUSSION

6.1 Conclusion

This study is mainly focused on applying Deep Learning with vehicle classifications. the proposed methods are Convolutional Neural Networks (CNN). They are experimented with two classifying tasks, i.e. vehicle type and vehicle color classification. The performance of those classifiers are then compared with both CNNs in the same classification in term of accuracy.

In this study, there are two experiment's setups designed for each task. In vehicle type classifying task, Non-pretrained weights InceptionV3 with augmented dataset achieved the highest mean accuracy (92.325%) with acceptable standard deviation (2.042). While Non-pretrained weights ResNet50 with augmented dataset is the best classifier in vehicle color classification which reached the highest mean accuracy (86.075%) with quite low standard deviation (1.493). Overall classifications' performance achieved the objective goal which is more than 85% accurate. The average accuracy of both classification is 89.2% which is higher than the goal by 4.2%. Furthermore, this research outperformed Saripan et al. [2] in both vehicle type and color classifications (79.82% and 69.29% respectively).

From the experiments, both best classifiers of each classification proved that augmented dataset improves the performance of ResNet50 and InceptionV3 models. In addition, even though the pretrained weights did work with Alexnet and VGG-19 models, they did not work well with InceptionV3 model and especially ResNet50 model. In other words, pretrained weights made ResNet50 to be overfitting because of how deep ResNet50 is. However, pretrained weights make the training process finish faster. In conclusion, pretrained weights is not suitable with the CNNs, e.g. ResNet and Inception that are designed to learn and to understand very deeply about the specific image input and the augmented dataset could improve the performance of CNN classifier. It is the trade-off in amount of training time consumed and the performance of classification depend on the expected objectives.

6.2 Discussions

From the author's observation, there are two major components that make the performance of prediction to be better. Firstly, the depth of the convolution neural network, if the depth of the network is deeper or more complex, it might perform better. Nowsaday, there are many CNN

structures that is deeper than the proposed methods, e.g. ResNet152, ResNext101, InceptionV4, InceptionResNetV2 and etc. In other words, deeper or more complex means more parameters which could represent many features or many ways of understanding the information of input images in the correct categories. Lastly, the numbers of related and useful images in dataset. This component is really important, it improves the performance considerably. More images give more the performance of classification. However, the numbers of images in each category should be balance as well.

For the future work, the CNN structures which are constructed later than 2016 will be conducted in the experiment in order to measure and compare the performance of prediction. Furthermore, more vehicle images should be gathered.

REFERENCES

[1] K. Saripan, C. Deachakul, and W. Moungmai, "Smart vehicle search in surveillance video," Bachelor Thesis, King Mongkut's Institute of Technology Ladkrabang, 2015.

[2] K. Saripan and C. Nuthong, "Tree-based vehicle classification system," in *2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, June 2017, pp. 439–442.

[3] R. F. Rachmadi and I. K. E. Purnama, "Vehicle color recognition using convolutional neural network," *CoRR*, vol. abs/1510.07391, 2015. [Online]. Available: http://arxiv.org/abs/1510.07391

[4] Y. Zhou, H. Nejati, T. Do, N. Cheung, and L. Cheah, "Image-based vehicle analysis using deep neural network: A systematic study," in *2016 IEEE International Conference on Digital Signal Processing (DSP)*, Oct 2016, pp. 276–280.

[5] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *CoRR*, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Neural Information Processing Systems*, vol. 25, 01 2012.

[7] B. Su, J. Shao, J. Zhou, X. Zhang, and L. Mei, "Vehicle color recognition in the surveillance with deep convolutional neural networks," 01 2015.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[9] M. Lin, Q. Chen, and S. Yan, "Network in network," *CoRR*, vol. abs/1312.4400, 2014.

[10] J. Patterson and Gibson, *Deep learning: a practitioners approach*. OReilly Media., 2017.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

APPENDICES

APPENDIX A Publication

February 23-25, 2019
SINGAPORE

ICCCS 2019

Proceedings of
2019 IEEE 4th International Conference on
Computer and Communication Systems

IEEE

ISBN: 978-1-7281-1321-0
IEEE Catalog Number: CFP19D48-USB

Proceedings of 2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS 2019)
ISBN: 978-1-7281-1321-0
IEEE Catalog Number: CFP19D48-USB

êèéñ & ìÈ– &«È^¾«sÈ˜±«s¤ ±«‘^¾^«•^ ±«
±ª»ÍÈ^¾ s«„ ±ªªÍ«˜•sÈ˜±« PÝÂÈ^ªÂ

P˜«’s»±¾^ Ø ^~¾Ís¾Ýêë êí÷ êèéñ
PÍ~ª˜ÂÂ˜±« ^s„¤˜«^ö 'HFHPEHU

ÜuFheY[[Z_d]i

$FFHSWHG SDSHUV ZLOO EH SXEOLVKHG LQ WKH FRQIHUHQFH SURFHHGLQJV ZKLFK ZLOO EH VXEPLWWHG IRU ,QGH[LQJ LQWR ,(((;SORUH VXEPLWWHG IRU &RQGHQVH LQ DQG FRSXV

Ü Jef_Yi

- $OJRULWKPV
- %LJ 'DWD
- &RPSXWHU $UFKLWHFWXUH
- 'DWD &RPSUHVVLRQ
- ,PDJH 3URFHVVLQJ
- 0RELOH &RPSXWLQJ
- +LJK 3HUIRUPDQFH &RPSXWLQJ
- $XWRQRPLF DQG 7UXVWHG &RPSXWLQJ
- 3DUDOOHO DQG 'LVWULEXWLQJ &RPSXWLQJ
- %LRPHGLFDO ,QIRUPDWLFV DQG &RPSXWDWLRQ
- 6RIWZDUH (QJLQHHULQJ DQG .QRZOHGJH (QJLQHHULQJ
- :LUHOHVV &RPPXQLFDWLRQV
- 1HWZRUN &RPPXQLFDWLRQ
- &RPPXQLFDWLRQV 7UDQVPLVVLRQ
- 1HWZRUN 6HFXULW\ DQG &U\SWRJUDSK\
- :LUHOHVV DQG 6HQVRU 'HYLFHV
- 5HPRWH 6HQVLQJ DQG *36
- 5) DQG 0LFURZDYH &RPPXQLFDWLRQ
- ,QIRUPDWLRQ DQG ,WV 7HFKQLFDO (GXFDWLRQ
- 6SHHFK DQG $XGLR 3URFHVVLQJ
- 6LJQDO ,PDJH DQG 9LGHR 3URFHVVLQJ
- 6LJQDO 'HWHFWLRQ DQG 3DUDPHWHU
- $UWLILFLDO ,QWHOOLJHQFH DQG 0DFKLQH /HDUQLQJ
- 5) 0LFURZDYH DQG PLOOLPHWHU FLUFXLW
- 7HFKQLTXHV RI /DVHU
- $QWHQQD DQG 3URSDJDWLRQ
- 5) DQG 0LFURZDYH GHYLFHV
- (OHFWURPDJQHWLF DQG 3KRWRQLFV
- 0LFURZDYH 7KHRU\ DQG 7HFKQLTXHV
- 9LUWXDO 5HDOLW\ DQG 9LVXDOL]DWLRQ
- 0RGXODWLRQ &RGLQJ DQG &KDQQHO $QDO\VLV
- ,QWHJUDWHG 2SWLFV DQG (OHFWUR RSWLFV 'HYLFHV

ÜuA[odej[If[Wa[hi

¼3URI 3HUU\ 6KXP
26$)HOORZ 63 ( )HOORZ
1DQ\DQJ 7HFKQRORJLFDO 8QLYHUVLW\
¼3URI *XX &KDQJ <DQJ ,((( )HOORZ
1DWLRQDO &KXQJ +VLQJ 8QLYHUVLW\
¼3URI <DQJ ;LDR ,(7)HOORZ
7KH 8QLYHUVLW\ RI $ODEDPD 86$

Üu>_ijeh

¼&&&6
.DQ\DNXPDUL ,QGLD _ 1RYHPEHU
3XEOLVKHU ,(((3UHVV ,6%1
3DSHUV RI ,&&&6 KDYH EHHQ LQGH[H
6FRSXV
¼&&&6
UDFRZ 3RODQG _ -XO\
3XEOLVKHU ,(((3UHVV ,6%1
3DSHUV RI ,&&&6 KDYH EHHQ LQGH[H
6FRSXV
¼&&&6
1DJR\D ,QVWLWXWH RI 7HFKQRORJ\ 1DJ
3XEOLVKHU ,(((3UHVV ,6%1
7KH FRQIHUHQFH SURFHHGLQJV RI ,&&&
;SORUH

Üu9ecc_jj[

&RQIHUHQFH &KDLUV
3URI <DQJ ;LDR 7KH 8QLYHUVLW\ RI $O
3URI *XX &KDQJ <DQJ 1DWLRQDO &KXQ
3URJUDP &KDLUV
3URI %R <DQJ 8QLYHUVLW\ RI (OHFWUR
&KLQD &KLQD
3URI 1REXR)XQDELNL 2ND\DPD 8QLYHU
3XEOLFLW\ &KDLU
$VVRF 3URI .U]\V]WRI .RV]HOD 3R]QDQ
3RODQG

ÜuIkXc_ii_

/RJ LQ WKH (OHFWURQLF 6XEPLVVLRQ
$Q\ TXHVWLRQV DERXW VXEPLVVLRQ
LFFFV#DFDGHPLF QHW

2019 IEEE 4th International Conference on Computer and Communication Systems

Vehicle Classification with Deep Learning

Watcharin Maungmai
International College
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
e-mail: 60610017@kmitl.ac.th

Chaiwat Nuthong
International College
King Mongkut's Institute of Technology Ladkrabang
Bangkok, Thailand
e-mail: chaiwat.nu@kmitl.ac.th

***Abstract*—Nowadays, there are many traffic surveillance systems which are installed in almost every city to record events and traffic. The surveillance system is used for various objectives, e.g. vehicles searching and real-time traffic monitoring, etc. For the searching purpose, the system can be used by policeman such as outlaw's vehicle identification in crime. Typically, the officers manually identify the vehicle in recorded video according to its appearances. Although the accuracy of this approach is good, it is time-consuming and inclined to faults due to human fatigue for long duration videos. Moreover, hiring employees is costly. Recently, there are several machine learning methods which can be applied to classify vehicles, e.g. Fuzzy Logic, Decision Tree, Adaboost, Random Forest, Neural Network, etc. Convolutional Neural Network (CNN) is also one of such methods. CNN is a type of Deep Learning which is in the category of the neural network. The method is very well-known in image recognition field at the present because of its performance. In the proposed vehicle classification, there are two vehicle characteristics, i.e. types and colors. Types consist of four classes while colors consist of seven classes. CNN is then used as to classify vehicle images. The experimental results show that CNN can achieve high performance in real-world applications.**

Keywords-vehicle classification; size classification; color classification; deep learning; convolutional neural network

I. INTRODUCTION

Nowadays, surveillance cameras are installed almost everywhere in cities. The main objectives of installing surveillance systems are real-time monitoring and events searching. In this paper, the authors focus only on events searching. For the searching objective, the surveillance system can be used by police officers. For example, in order to search for specific vehicle. In general, the officers require the information of vehicle's characteristics, e.g. vehicle's color, vehicle's type as a clue for vehicle identification. The officers often spend a lot of time monitoring recorded videos by themselves. Typically, searching time is usually more than video duration and they have to repeat the searching task again several times. In addition, the officers might make some mistakes with their weariness after a period of searching.

In order to solve such problems, vehicle classification can be utilized in order to assist the vehicle searching. Various methods are applied in vehicle classification at present. K. Ying et al. [1] proposed a decision tree as a classifier. In their experiment, feature combinations are used in order to reduce memory and computational time. However, the combinations of four or more features couldn't make classification accuracy increases. R. Feris et al. [2] constructed a system which could search for vehicles in surveillance videos. They proposed a new classifier named Motionlet. The classifier was a detector based on Adaboost learning [3]. The main task of Motionlet was categorizing twelve different direction of vehicles. As a result, they could achieve 87% accuracy rate. S. B. Changalasetty et al. [4] used an artificial neural network as a classifier which classified vehicles into two categories, i.e. big and small. Their results achieved more than 90% accuracy rate. P.O. Gislason et al. [5] compared the performance of classification among various classifiers, i.e. regression tree, bagging, boosting, and random forest. In their experiment, all methods gave results which were comparable to each other. However, random forest was chosen since two reasons, i.e. shorter learning time compared to other methods and required no guidance. Saripan et al. [6], [7] proposed the vehicle search system. The system used a surveillance video as input and allowed user to select vehicle characteristics in order to search for specific vehicles. The authors also proposed Tree-based vehicle classification in order to categorize vehicles from the video. In their experiment, the classification worked well when combined with the system.

In recent decades, there is another method call Deep Learning [8] which can be used in classification task. Deep learning is a neural network with more than two hidden layers. Following are some of the facets in evolution of neural network:

- More neurons than previous networks
- More complex ways of connecting layers/neurons in neural networks
- Explosion in the amount of computing power available to train
- Automatic feature extraction

There are many types of deep learning, e.g. unsupervised pretrained networks, convolution neural networks (CNN), recurrent neural networks, recursive neural network, etc. Convolutional neural networks are the most popular neural network in deep learning. The main characteristic of these networks is convolution, which is designed to learn higher features in the data. The networks are well suited to object recognition with images and consistently top classifier in image classification competitions. Krizhevsky et al. [9] is the most popular CNN which won the ILSVRC (ImageNet Large Scale Visual Recognition Challenge) 2012. The

978-1-7281-1321-0/19/$31.00 ©2019 IEEE

efficacy of CNNs in image recognition is one of the main reasons why the world recognizes the power of deep learning.

This paper proposes a convolutional neural network framework to circumvent the previously mentioned problems in vehicle searching in surveillance videos. This work focuses mainly on the performance of vehicle classification modules [6], [7] which are vehicle type classification and vehicle color classification. The detail is explained in related work and proposed method sections.

The remainder of this paper is organized as the follows. Section II explains the related works. It is then followed by Section III which explains proposed method in the details included CNN structure. Section IV shows experimental result which consist of experiment setup and accuracy of the result. Finally, Section V concludes the experimental results and discussion about future works.

II. Related Work

There are several researches which used CNN as a classifier in vehicle color classification [10]-[12]. Chen et al. [13] proposed feature context as an approach to identity the color of the vehicle. They applied the classification with their dataset which contains 15,601 vehicle images with 8 classes of vehicle color. They could achieve 90.68% accuracy in the classification. The authors in [10], [12] applied CNN with Chen's dataset [13]. Their results showed 94.47% and 94.6% accuracy rate, respectively. Su et al. [11] proposed new CNN structure named Colornet which achieved the highest accuracy in their vehicle color classification experiment of 95.74%. The structure outperformed Alexnet [9] and GoogleNet [14].

Another work by Zhou et al. [15] proposed deep neural network or deep learning approaches for vehicle detection and vehicle classification. In detection, they used YOLO [16] as a detection model. Alexnet [9] was used as classification approaches. In classification modules, there are four kinds of classification, i.e. passenger vs other, cars vs vans, sedans vs taxis, and sedans vs vans vs taxis. After applying both structures, they were fine-tuned to be suitable with the public dataset which provided in [17]. The experimental results showed the accuracy of more than 90%.

Saripan et al. [7] proposed a tree-based vehicle classification system which required a surveillance video and vehicle's characteristics as inputs. This work is proposed based on search system. The system consists of three modules, i.e. feature extraction, classification, and search manager. Fig. 1 shows the overview of the system. The video is used as an input at the beginning in feature extraction module. The module crops vehicle images from the video systematically and extracts the features of those images which are listed in the Table I. These extracted feature data is then sent to the classification module. Fig. 2 shows the overview of the classification module. There are two classification, i.e. type and color. In type classification, four classes are categorized, i.e. small, medium, large and unknown. There are seven classes in color classification, i.e. black, white, blue, green, yellow, red, and unknown. Note that, both classifications consist of unknown class. This class

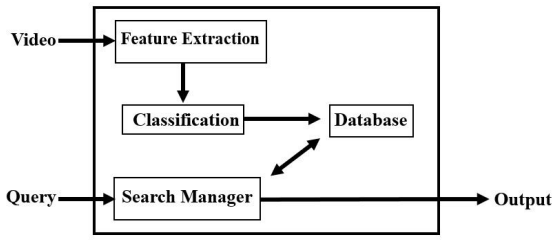

Figure 1. System overview.

TABLE I. Input Features in Vehicle Type and Vehicle Color Classification Task

Type Input Features	Color Input Features
Bounding box • X coordinate from top-left corner • Y coordinate from top-left corner • Width • Height Ratio of vehicle over background	Color in HSV space • Hue • Saturation • Value

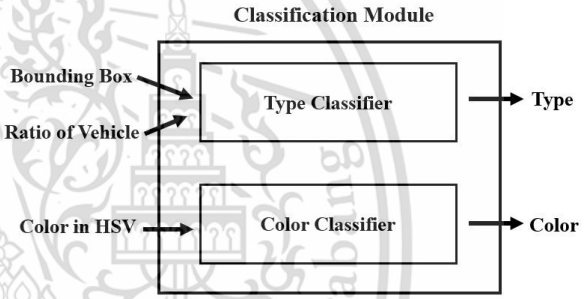

Figure 2. Classification module overview.

TABLE II. Output Classes in Vehicle Type and Vehicle Color Classification Task

Type Output Classes	Color Output Classes
Small Medium Large Unknown	Black White Red Blue Yellow Green Unknown

contains the vehicle with ambiguous characteristics and irrelevant colors, e.g. overlapped vehicles, brown color, etc. Overall possible target classes in type and color classification are shown in Table II. The results from classification module are sent to the search manager module. This module then stores and filters the results according to the given query commands.

In this paper, the main objective of the proposed method is to improve the accuracy of vehicle type and vehicle color classification which are previously mentioned. The classifier in this work is selected to be convolutional neural network with two convolution layers. The CNN is chosen because of its performance in image recognition. The vehicle images of

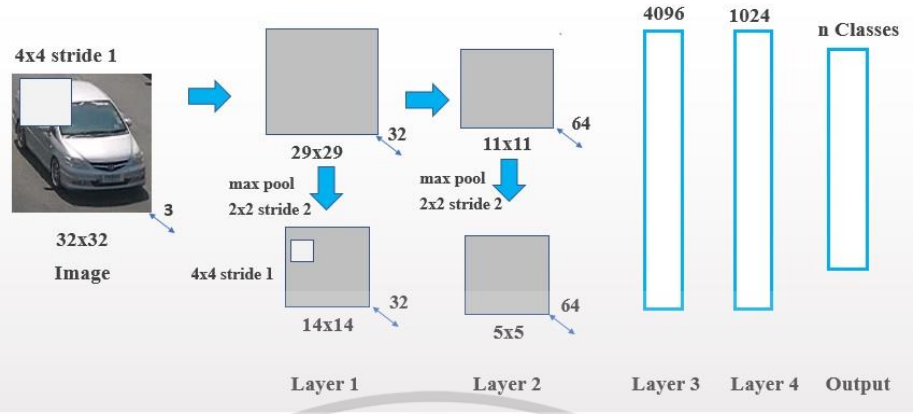

Figure 3. Architecture of our 5 layers convolution neural network model. A resized image 32 by 32 is presented as the input. This is convolved with 32 different 1st layer filters, each of size 4 by 4, using a stride of 1 in both width and height. The results are feature maps which then: (i) passed through a rectified linear fuction(ReLU) and (ii) The results from ReLU function are sent to pooling layer which using max pooling with 2 by 2 size and stride of 2. Similar operations are repeated in layer 2. The next two layers are fully connected, taking features from the last convolution layer as input in vector form. The final layer is a softmax function, n being the number of classes.

dataset provided in [6] is fed into CNN structure. More detail will be explained in following section.

III. Proposed Method

This work consists of a CNN structures which are used as classifiers in both vehicle type classification and vehicle color classification. The proposed method requires only one input which is a vehicle image fed into the system. The outputs of classifications are exactly the same as listed in Table II mentioned previously.

A. Convolutional Neural Network

Convolutional Neural Network is a kind of feed forward artificial neural network, it is quite a similar to standard neural network. The neurons in the network have learnable weights and biases. Every neuron receives inputs and performs some operations. There are three major layers in CNN, i.e. convolution layer, pooling layer, and fully connected layer. Convolution layer will calculate the output of neurons that are connected to local regions within the input, each computes a dot product between the weights and biases. Pooling layer is used in order to reduce the feature maps' size. It means that the parameters will be reduced too, the computation time is than faster. In general, max pooling is used in CNN. In fully connected layer, each neuron in this layer is connected to previous layer neurons. The layers are fully connected as in the same manner as in a common neural network.

B. The CNN Architecture

The proposed CNN architectures contain the following characteristics which are shown in Table III. There are two convolution layers, i.e. 1st and 2nd layer. In the structure, pooling layers are set at the positions after each convolution layer which is already applied with activation function. The activation function is used to operate after the process of convolution and fully connected, although the last layer is not applied the function. The 3rd, 4th and 5th layer are fully connected layers. Dropout is a common method which can

TABLE III. Components of the Proposed CNN

Input	A 32×32 resized image
Convolution layers	1st layer and 2nd layer
Pooling layers	2 pooling layers
Activation functions	4 ReLU functions
Fully connected layers	3rd layer, 4th layer and 5th layer
Dropout	1 dropout
Output	The number of possible classes

Figure 4. Sample vehicle images from Saripan dataset [6].

be used to avoid overfitting. Output or predictor is the final layer. The number of neurons in this layer is equal to the number of possible classes. Moreover, the outputs are predicted classes with probability score in range 0 to 1.

Fig. 3 shows the structure of the CNN. Firstly, the original vehicle image is resized into 32×32 pixels by using resize function in Tensorflow. The resized image is fed to the 1st convolutional layer with 32 filters of size 4×4×3 and stride of one pixel. The output of the first convolution layer is modeled by ReLU. Max pooling reduces the size of the feature map outputs with kernel size of 2×2 and the stride of two pixels. Then the output is passed to the 2nd convolutional layer and the same operations are repeated. After that, the output of 2nd convolution layer is converted into vector form and then fed into fully connected layer. In the fully

connected layers, there are dot product operation and ReLU operation, respectively. The dropout is added at the position after the 4th layer in order to prevent the overfitting in training networks. The final layer is a predictor which is softmax. In addition, a number of neurons are equal to a number of possible classes. In this work, the last layer of vehicle type classification contains four neurons and there are seven neurons at last layer of vehicle color classification. (Fig. 4)

IV. Experimental Result

In the experiment, the proposed method is evaluated by comparing with Saripan et al. [6], [7] and densely deep neural network. This section consists of three parts, i.e. dataset, environments and evaluation results.

A. Dataset

In this work, vehicle images from dataset [6] are being used. They are systematically extracted by a system. The following are characteristics of the dataset:

- Video duration: 763 seconds
- Resolution: 640×480 pixels
- Frame rate: 25 fps
- Extracted vehicle images: 914 images

The dataset is separated into two parts, i.e. training part and testing part which are 75% (686 images) and 25% (228 images) of dataset, respectively.

B. Environments

All data preparation, analyzing process, and computation are performed in Jupyter Notebook. The CNN structures are implemented by using Tensorflow. A workstation used in conducting experiments is equipped as follows:

- Intel (R) Core (TM) i5-4690 CPU Processor running at 3.50 GHz clock frequency
- 16 GB of DDR3 memory running at 799.5 MHz
- NVIDIA GeForce GTX 750 Ti with 2048 MB GDDR5 memory

The hyperparameters of both CNN structures are shown in Table IV. These parameters are the main components which affect to the performance of the CNN models. In the experiment, the classifier is Softmax. Dropout rate is 40%.

The learning rate of Adam optimizer is initialized to be 0.001 at the beginning. Batch size of 2 is calculated from the greatest common factor of two numbers, i.e. the number of training samples and the number of test samples. The number of epochs is 5,000.

C. Evaluation Results

In order to compare the performance between proposed method and densely deep neural network. In the densely DNNs, the number of epochs is set to be 5,000. The hidden layers of densely DNNs are 15,12,15,10 and the output layer's neurons are equal to the number of possible classes. Moreover, the DNNs is fed with the features which shown in Table I. The features are also fed into the system as mentioned in [6], [7]. The experiments are comparable since those features are extracted from the vehicle images which will be fed into the proposed method. The results of vehicle type classification and vehicle color classification are shown in Table V and Table VI, respectively. In type classification experiment, the proposed method achieves more than 80% and outperforms the random forest by 1.8% accuracy. However, the proposed method achieves 70.09% in color classification which is more than the adaboost only 0.8% accuracy. In this work, the proposed method has been run 10 times. One vehicle color and vehicle type classification result

TABLE IV. Hyperparameters of the Proposed CNN

Convolution layers	4×4 filter with stride of 1
Fully connected layers	4096, 1024 and 4 or 7 (a number of possible classes) neurons, respectively
Pooling layers	Max pooling with 2×2 filter and stride of 2
Classifier	Softmax
Dropout rate	0.4
Batch size	2
Optimizer	Adam with learning rate 0.001
A number of epochs	5000

TABLE V. Vehicle Type Classification Results

Method	Accuracy (%)
Decision tree [6]	79.38
Random forest [7]	79.82
DNN(Densely)	79.31
CNN	**81.62**

TABLE VI. Vehicle Color Classification Results

Method	Accuracy (%)
Fuzzy logic [6]	62.72
Adaboost [7]	69.29
DNN(Densely)	65.52
CNN	**70.09**

TABLE VII. Confusion Matrix of Vehicle Color Classification

		Classified as						
		R	***Y***	***G***	***BL***	***BK***	***W***	***U***
Actual	***R***	0	0	0	0	5	0	2
	Y	0	3	0	0	0	0	0
	G	0	0	1	0	0	0	0
	BL	0	0	0	0	1	0	0
	BK	0	0	0	2	31	0	5
	W	0	0	0	0	0	70	23
	U	3	2	0	1	4	10	65

The abbreviations for color are as follows, Red (R), Yellow (Y), Green (G), Blue (BL),
Black (BK), White (W), Unknown (U)

TABLE VIII. Confusion Matrix of Vehicle Type Classification

		Classified as			
		Small	***Medium***	***Large***	***Unknown***
Actual	***Small***	60	9	0	2
	Medium	17	89	2	7
	Large	0	0	13	0
	Unknown	1	0	0	28

TABLE IX. Overall Vehicle Classification with the Proposed CNN Method

Classification	Accuracy (%)			
	Max	***Min***	***Mean***	***SD***
Type	84.65	78.07	81.62	2.597
Color	75.44	67.11	70.09	2.27

of those experiments have been shown as confusion matrices in Table VII and Table VIII, respectively. In conclusion, the experiment results are shown in Table IX, these experiments demonstrate that CNN could achieve better results, 84.65% in type and 75.44% in color classification. However, the results are not quite stable with standard deviation 2.597 and 2.27, respectively. From the results, the authors aim to search for CNN's suitable hyperparameters which could make the performance of the classifications be more accurate and stable.

V. CONCLUSION AND DISSCUSION

In this paper, CNN is proposed as a type and color classifiers to classify vehicle characteristics from vehicle image which systematically cropped by machine. The experiment's results show that CNN outperforms other methods in type classification. However, CNN improves a little accuracy in color classification. For the future work, improving the accuracy of color classification will be the major goal. Furthermore, many aspects should be explored and experimented, e.g. different input image size also, deeper CNN structure such as Alexnet structure which many researches applied and fine-tuned the structure and different hyperparameters.

ACKNOWLEDGMENT

The authors would like to thank International College, King Mongkut's Institute of Technology Ladkrabang for providing us resources and supports.

REFERENCES

[1] K. Ying, A. Ameri, A. Truvedi, D. Ravindra, D. Patel, and M. Mozumdar, "Decision tree-based machine learning algorithm for in-node vehicle classification," 2015 IEEE Green Energy and Systems Conference (IGESC), Long Beach, CA, 2015, pp.71-76. DOI:10.1109/IGESC.2015.7359454

[2] R. Feris, B. Siddiquie, Y. Zhai, J. Petterson, L. Brown, and S. Pankanti, "Attribute-based vehicle search in crowded surveillance videos," In Proceedings of the 1st ACM International Conference on Multimedia Retrival(ICMR '11). New York, 2011.

[3] P. Viola and M. Jones, "Robust Real-time Object Detection," International Journal of Computer Vision, 2001.

[4] S.B. Changalasetty, A.S. Badawy, L.S. Thota, and W. Ghribi, "Moving vehicles classification in WEKA," 2015 International Conference on Advanced Computing and Communication Systems, Coimbatore, 2015, pp. 1-6 DOI: 10.1109/ICACCS.2015.7324081

[5] P.O. Gislason, J.A. Benediktsson, and J.R. Sveinsson, "Random forests for land cover classification, "Pattern Recognition Letters, vol. 27, no. 4, pp. 294300, Mar. 2006

[6] K. Saripan, C. Deachakul, and W. Moungmai, "Smart vehicle search in surveillance video," 2015

[7] K. Saripan and C. Nuthong, "Tree-based vehicle classification system," 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, 2017, pp. 439-442.

[8] Patterson, J. and Gibson, A. (2017). Deep learning: a practitioner's approach. Sebastopol: O'Reilly Media.

[9] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks." In Advances in neural information processing systems, 2012, pp. 1097–1105.

[10] R. F. Rachmadi, I. K. E. Purnama, M. H. Purnomo, "Vehicle Color Recognition using Convolutional Neural Network." arXiv preprintarXiv:1510.07391

[11] B. Su, J. Shao, J. Zhou, X. Zhang and L. Mei, "Vehicle Color Recognition in The Surveillance with Deep Convolutional Neural Networks," 2015 Joint International Mechanical, Electronic and Information Technology Conference, Chongqing, 2015

[12] C. Hu, X. Bai, L. Qi, P. Chen, G. Xue, and L. Mei, "Vehicle Color Recognition With Spatial Pyramid Deep Learning.", IEEE Transactions on Intelligent Transportation Systems, 2015

[13] P. Chen, X. Bai, and W. Liu, "Vehicle Color Recognition on Urban Road by Feature Context.", IEEE Transactions on Intelligent Transportation Systems, 2014

[14] C. Szegedy, W. Liu, Y. Jia, and A. Rabinovich, "Going deeper with convolutions.", In arXiv:1409.4842

[15] Y. Zhou, H. Nejati, T. T. Do, N. M. Cheung and L. Cheah, "Image-based vehicle analysis using deep neural network: A systematic study," 2016 IEEE International Conference on Digital Signal Processing (DSP), Beijing, 2016, pp. 276-280.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," arXiv preprintarXiv:1506.02640, 2015.

[17] X. Ma and W.E.L. Grimson, "Edge-based rich representation for vehicle classification," in Tenth IEEE International Conference on Computer Vision (ICCV). IEEE, 2005, vol. 2, pp. 1185-1192

BIOGRAPHY

Personal Information

Name	Watcharin Maungmai
Sex	Male
Nationality	Thai
Date of Birth	9 March, 1993
Place of Birth	Suanluang District, Bangkok City, Thailand

Education

Bachelor degree

Project	Smart Vehicle Search in Surveillance Video
Field of Study	Software Engineering
Duration	2012-2016
Department	International College
University	King Mongkut's Institute of Technology Ladkrabang, BangKok, Thailand

Master degree

Thesis	Vehicle Classification with Deep Learning
Field of Study	Computational Intelligence System
Duration	2017-2019
College	International College
University	King Mongkut's Institute of Technology Ladkrabang, BangKok, Thailand

Research Interests

Object Recognition, and Deep Learning