APPLICATION OF ENSEMBLE CLASSIFIERS IN SMART VEHICLE SEARCH SYSTEM

KIATKACHORN SARIPAN

A THESIS REPORT SUBMITTED IN PARTIAL FULLFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN COMPUTING IN ENGINEERING SYSTEM
INTERNATIONAL COLLEGE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2016
KMITL-2016-IC-M-11-01

Abstract

Traffic surveillance systems give recorded videos which can be used to search for specific vehicle of interest. The vehicle search system consists of three modules, i.e. feature extraction, classification, and search manager, has been proposed. The feature extraction module is used for image and video processing. It extracts the desired features to be used further in the classification module. The classification module uses these features and results in pre-defined vehicle classes. The classification results are stored in the search manager module for further filtering according to user's query command.

However, the overall performance of the proposed system can be improved further. This work focuses on the performance improvement of the classification module. There are two features designed to be used in the proposed classification module, i.e. types and colors. Vehicles are classified into four classes of type and seven classes of color. The previous proposed algorithm is a Decision tree which gives incorrect classification result for some vehicle records. This can be improved further by applying ensemble approach method. Several ensemble tree based algorithms are applied to the dataset, e.g. Adaboost, Bagging, Random forest. The experimental results show that all the algorithms are comparable. However, the highest accuracy for type and color classification are obtained by using Decision tree and Bagged Decision tree, respectively.

Acknowledgement

This thesis couldn't be completed if I don't have very great advices from my advisor, Asst.Prof.Dr. Chaiwat Nuthong. Thank you for knowledge, suggestions, and comments from Dr. Isara Anantavrasilp, Dr. Ukrit Watchareeruetai, Dr. Natthapong Jungteerapanich, and Assoc.Prof.Dr. Veera Booning. Including all my colleagues for the support and help.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

There are traffic surveillance systems installed in major cities. Most of them use visual method to capture traffic and events. The collected information can be used for two main purposes, i.e. monitoring and searching. In this work, monitoring refers to viewing real-time events in video streaming while searching means scanning for specific events in the surveillance videos. This project focuses on searching task.

Searching in a surviellance video can be done for many reasons, one of them is to support police officers in crimes investigation related to transportation, e.g. hit-and-run, stolen vehicles tracking, suspected vehicles searching, etc. In traffic crimes investigation, police officers might want to have evidences of vehicles recorded in the traffic surveillance system. This is because a suspect is usually caught on cameras when he escapes from the scene by using a vehicle. However, in case of lacking of an automated searching system for videos, the police officers have to perform the task manually. By using this traditional approach, it usually requires more time than the video's period. In addition, manually search also affects the accuracy depending on video's duration. This is because police officers might get tired after watching such a long video.

It can be seen that searching manually leads to two main problems, i.e. long searching time and low accuracy in case of long video's duration due to human fatigue. In order to solve these problems, automated searching system is proposed. Smart Vehicle Search in Surveillance Videos [8] is one of them. This system consists of three modules, feature extraction, classification, and search manager. According to their conclusion, the system can search for a specific vehicle automatically. However, the classification module can be improved further. This project focuses on searching for a method which can improve the performance of the classification module. The improvement shall be in terms of training time and searching accuracy.

1.1 Problem Descriptions

Traffic surveillance systems are used to monitor and search for traffic information. Officers need to extract information from the videos based on the evidence for each case. Currently, they have to search or monitor it manually by using their visual perception. This approach is time consuming. One of the reasons is that there is no index for each vehicle. In addition, in case of a long period video, this approach may lead to poor accuracy's result due to the officers' fatigue.

Several systems were proposed. One of them was designed to solve these problems. The system consists of three modules, i.e. feature extraction, classification, and search manager. The feature extraction module processes input video and sends the result to the classification module. The classification module categorizes vehicle into classes and stores the result in the search manager module. The search manager module filters the result according to the query command. Although, the system can be used to solve the problem, the accuracy of the system is not satisfied. This might lead to misclassification and cause some evidences being missed. Therefore, the system's accuracy needs to be improved. In order to acheive the goal, there are various approaches available, e.g. introducing new features to the feature extraction module, introducing new classification method, etc. In this work, the author focuses on improving the classification module.

1.2 Objectives

From the system proposed by Saripan et al. [8], there was a goal, overall system's accuracy, which was not satisfied due to the performance of the classificaion module. In order to improve the performance of the classification module, some goals are needed to be satisfied. The following list shows the goals of this work.

- The accuracy of each classifier is expected to be more than 80%.
- Training time is expected to be less than 0.5 seconds per 1,000 instances.

1.3 Proposed System

From the work proposed by Saripan et al. [8], manually search is time consuming and might be inaccurate due to human fatigue. It is found that the classificaion module's accuracy can be improved further. Therefore, in this work, the author focuses on improving the classificaion module's accuracy to satisfy the goals mentioned in the previous section. In addition, accuracy of the system shall be higher than the system proposed by [8].

The system proposed by [8] consists of three module, i.e. feature extraction module, classification module, and search manager module. This is illustrated in Figure 1.1. The system requires two inputs, i.e. location of the video and a query command. Location of the video is a path to the video file while query command refers to the characteristics of the searched vehicle.

The searching process begins with feature extraction module. It extracts characteristics of each vehicle in the video and sends this information to classification module. The information is classified by classification module. The obtained results are type and color of vehicles. Then, search manager module stores the classified data in database. This module shows the result according to the given query command.

The classification module consists of two classifiers inside, i.e. type and color classifier. Bounding box and detected vehicle foreground ratio are used by the type classifier to determine vehicle's type while color in HSV color space is used to determine vehicle's color. Figure 1.2 shows the the overview of the classification module.

Figure 1.1: Proposed smart vehicle search system overview

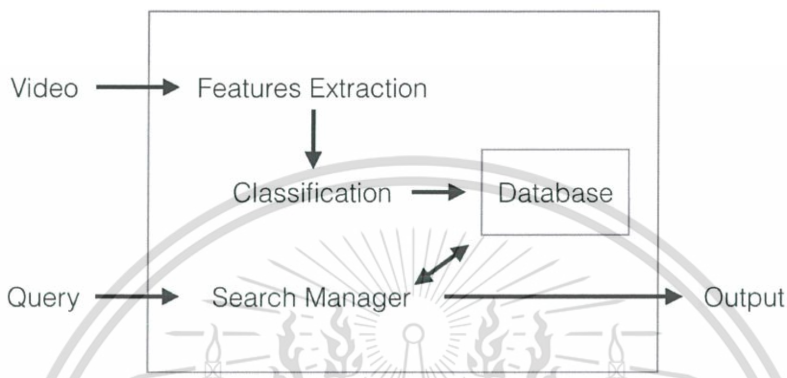
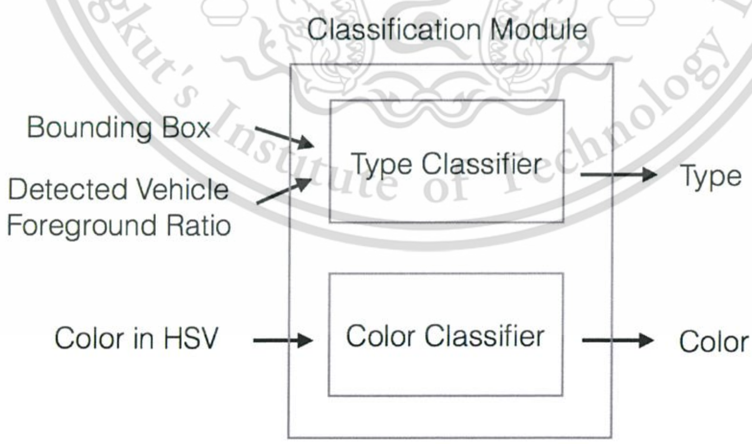


Figure 1.2: Classification module overview



1.4 Scope of Work

In this work, the author focuses on the improvement of the classification module. There are several approaches to fulfill the goals, e.g. feature extraction and selection, enhance the performance of the classifiers, etc. For feature extraction and selection, the methods involve preprocessing on the input feature vector, introducing new features, selecting the features which affect the classification accuracy. Apart from feature extraction and selection, enhancing the performance of the classifiers can also be done. This can be achieved by applying new approaches. This work focuses on the enhancement of the classifiers performance, in particular, ensemble enthods, i.e. Boosting, Bagging, and Random forest. The dataset is the same as [8]. The results are compared among the ensemble methods and Decision tree, obtained from [8].

1.5 Thesis Structure

This thesis contains five chapters. Chapter 2 discusses the concerning literature review and shows the concepts related to this project. Possible approches which can improve the classification module will be explained in chapter 3. Chapter 4 presents the experimental setup and results. In the experiment, dataset is split into two sets, i.e. training and testing set. Classifiers are contructed using the training set while the testing set is used to determine the performance of the model. Moreover, the results are also compared in this chapter. The conclusion and discussion of this work are shown in chapter 5. In addition, this chapter also includes future works.

Chapter 2

Literature Review

This project aims to solve the original problems concerning manually search and improve the accuracy of the system. From the work of Saripan et al. [8], the improvement can be done in the feature extraction module or the classification module. In the the feature extraction module, additional procedures could be implemented to obtain new features while new classifiers could be introduced to the classification module. From the experimental results in [8], it was shown that the classification module achieved lower accuracy compared to the feature extraction module. The details of literature review can be found in the following section.

2.1 Attribute-based Vehicle Search in Crowded Surveillance Videos

Feris et al. [2] presented a framework to search for vehicles based on the given attributes. The system could be split into three main parts, i.e. analytic engine, back-end database, and search interface. The high-level overview of the system is shown in Figure 2.1. Searching process was done by extracting information from input video, ingesting the data into database, and searching in the database using the given attributes. The result of the search included timestamps, estimated length of the vehicle in the output image, and snapshot of the vehicle from the video frame. Some of the results are shown in Figure 2.2

Figure 2.1: High-level system architecture

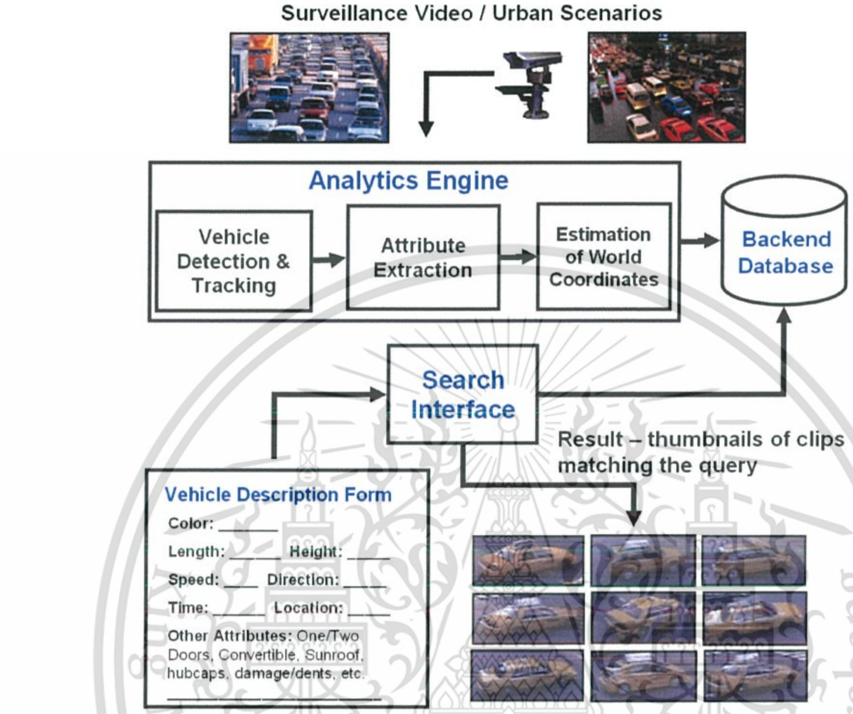
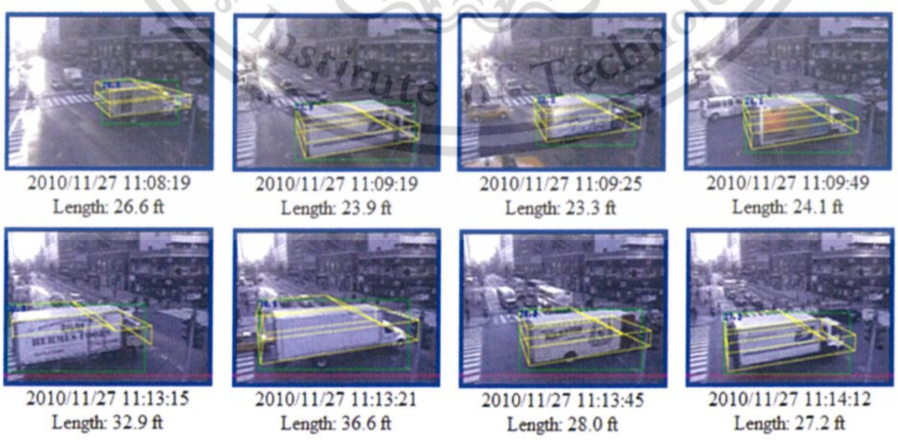


Figure 2.2: Some of sample searching results for vehicles with length larger than 18 [ft]



The attributes extraction was a main focus in this paper. First, the system needed to identify whether there was a vehicle in the given image. The authors proposed a classifier called a Motionlet detector. If there was a vehicle in the given image, the Motionlet detector would detect it, otherwise, it resulted in no vehicle detected. It should be noted that one Motionlet detector could detect a vehicle in only one direction. In their experiments, 12 Motionlet detectors were implemented. In case the vehicle was detected, its attributes, e.g. width, length, height, etc., would be extracted. These attributes were extracted using calibration matrix. This method gave the vehicle's dimension in real-world coordinate. Note that, the calibration matrix had to be configured manually before the system could operate. Another important attribute, a dominant color of the vehicle was extracted by quantizing the angular cutoff of Hue component in HSL color space into four colors, i.e. Red, Green, Blue, and Yellow. In case the lightness or saturation lied above or below the horizontal mid-plane the color would be relabeled to White or Black, respectively.

They setup experiments using the same dataset and applied three different methods, i.e. baseline method [6, 9], Motionlet, and Motionlet with large-scale feature selection. The comparison showed that the Motionlet with large-scale feature selection was the best method.

2.2 A Comparative Study of Decision Tree ID3 and C4.5

Hssina et al. [5] explained a Decision tree model in detail. There are many algorithms which can be used to construct the Decision tree model, e.g. Classification and Regression Tree (CART), ID3, C4.5, etc.. ID3 and C4.5 are two main focused algorithms in this paper. The authors showed the differences and compared the performance between these two methods.

Decision tree model building algorithm was first proposed in 1979 by Quinlan. This algorithm is known as ID3. It constructs a Decision tree model based on the information theory using Shannon entropy and information gain. Shannon entropy can be used to determine the uncertainty of the given data. This value is required further to compute Information gain.

The Information gain itself defines the amount of information contained in the given data. The explanation for these two concepts will be discussed in the following chapter.

In order to construct a Decision tree model using ID3 algorithm, Information gain is used to select the best attribute to be tested on each node. The training records are removed according to the test result. The process iterates until there is only one class in the same branch or there is no other attributes to separate the classes. An example of final tree which uses ID3 algorithm is shown in Figure 2.3. This Decision tree is constructed based on the dataset in [5].

Since ID3 algorithm handles only discrete values and is overly sensitive to the features with large numbers of values, it has limitation in applying to some datasets, for examples dataset with continuous value or large number of values. In order to circumvent these problems, C4.5 algorithm was proposed after the ID3 algorithm in 1993 by Quinlan. It is an improved version of ID3. The improvement of the algorithm allows it to be able to handle unknown values, continuous data, and tree pruning.

Figure 2.3: Final tree model based on ID3 algorithm

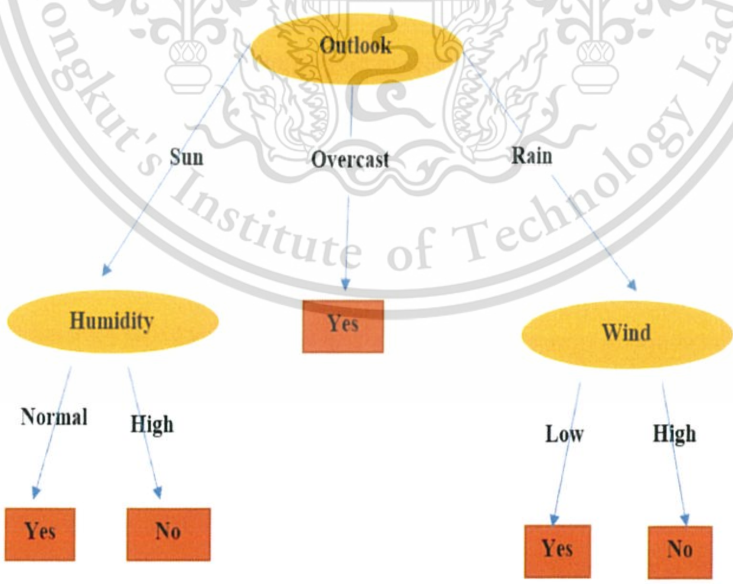


Table 2.1: Accuracy comparison between ID3 and C4.5 algorithm [5]

Number of samples	Accuracy	
	ID3 (%)	C4.5 (%)
14	94.15	96.20
24	78.47	83.52
35	82.2	84.12

Table 2.2: Execution time comparison between ID3 and C4.5 algorithm [5]

Number of samples	Execution time	
	ID3 (s)	C4.5 (s)
14	0.2150	0.0015
24	0.3200	0.1700
35	0.3900	0.2300

In conclusion, the authors compared the two methods in aspect of accuracy and execution time. The experimental results, Table 2.1 and Table 2.2, show that the C4.5 algorithm was able to construct a Decision tree model which gave higher accuracy with less execution time.

2.3 Random Forests For Land Cover Classification

Gislason et al. [3] investigated several classification methods to find the best method for multisource remote sensing and geographic data classification. The authors obtained dataset from four sources shown in the following list.

- Landsat Multispectral Scanner (MSS) data (four spectral data channels).
- Elevation data (in 10 m contour intervals, one data channel).
- Slope data ($0° - 180°$ in $1°$ increments, one data channel).
- Aspect data ($1° - 180°$ in $1°$ increments, one data channel).

The data could be categorized into ten classes, one of them was water and others were forest types. The authors split half of the dataset into

Table 2.3: Comparison of Random forest to Boosting and Bagging with various based classifiers in term of overall test set accuracy

Method	Overall accuracy(%)
CART (not ensemble)	78.3
Bagging (Decision table)	82.5
Bagging (J4.8)	81.7
Bagging (1R)	73.6
Boosting (Decision table)	83.8
Boosting (J4.8)	81.5
Boosting (1R)	85.3
Random forest	82.8

training and another half into testing set. These two datasets were used to construct and evaluate the performance of each classifier. Classification and regression tree, Bagging, Boosting, and Random forest were experimented. Accuracy of each classifier is shown in Table 2.3.

The result shows that all the methods were comparable in term of accuracy. However, the authors chose Random forest for their work because it could be trained much faster. Furthermore, it did not overfit the problem or require guidance.

2.4 Moving Vehicle Classification in WEKA

Changalasetty et al. [1] proposed a vehicle classification system which could classify a vehicle into two classes, i.e. big and small. In order to categorize vehicles, there were two steps, i.e. extracting information from the input video and categorize each vehicle into class.

The system started with extracting vehicles' information from the input video. Image processing techniques were applied on the input video to extract vehicles' information using LabVIEW[1] as an vision assistant. The

[1]Laboratory Virtual Instrument Engineering Workbench(LabVIEW) is a software which commonly used for data acquisition, instrument control, and industrial automation.

extracted vehicle's attributes are shown in the following list.

- Frame-id
- Vehicle-id
- Bounded-width
- Bounded-height
- Perimeter
- Particle & hole-area

The extracted information was used to determine the class of each vehicle, big or small. Neural Network was used as a classifier. It is a supervised machine learning algorithm. The structure is a directed-acyclic-graph mimic from human brain. The graph consists of two components, i.e. edges and vertices. Each edge has its own weight. This weight is used to multiply the input and send the result to next vertex. Each vertex contains an activation function which can compute the output from the input. Figure 2.4 shows example of Neural Network.

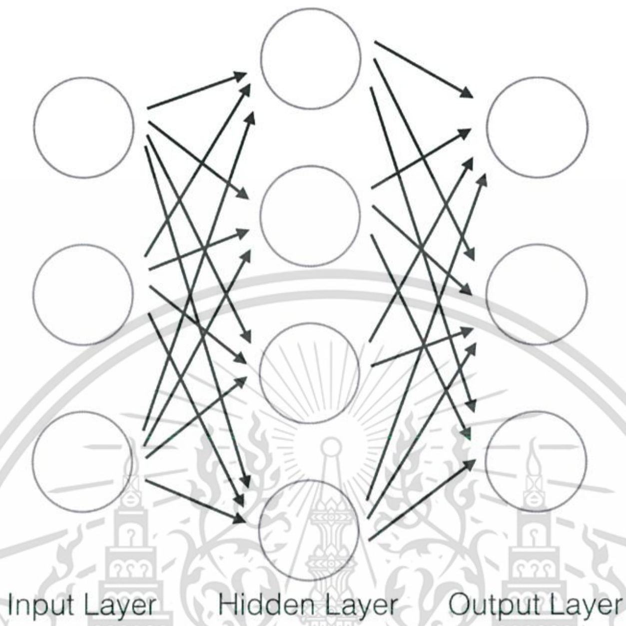
The model can be built by using backpropagation algorithm. This algorithm uses two processes in each iteration. The first step is to give the model an input vector and calculate for error. Then, the weight on each edge is adjusted to reduce the error. The process repeats until the error reduces to the satisfied value or reaches a defined number of iterations.

In their experiment, a video of two minutes was used as the dataset. The training set was randomly picked and used to build a model. The testing set was then applied to the obtained model. The result showed that the the model can be used to classify vehicle with over 90% accuracy rate.

2.5 Vehicle Classification Based on the Fusion of Deep Network Features and Traditional Features

Qian et al. [7] proposed an algorithm to classify a vehicle. Their algorithm was able to classify a vehicle into six categories, i.e. motorcycle, car, SUV,

Figure 2.4: Example of Neural Network

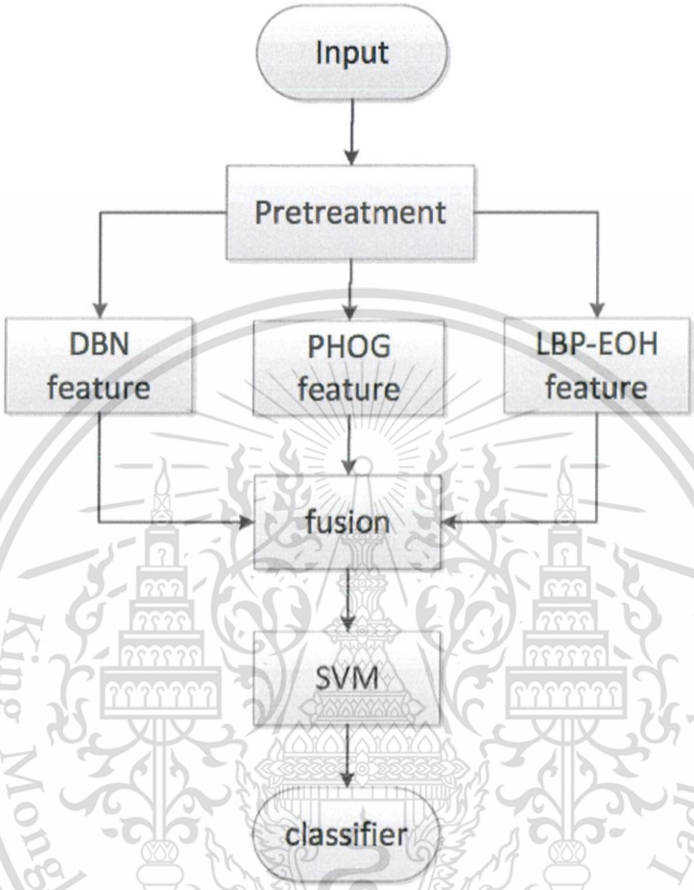


van, bus, and truck.

There were four steps in their algorithm, i.e. preprocessing, feature extraction, feature fusion, and recognition and classification. The preprocessing step was done to prepare the image for the feature extraction step. In this step, the image was duplicated into three identical images. Each image was prepared according to the feature which would be extracted in the feature extraction step. The features which would be extracted were Deep Belief Network (DBN), Pyramid Gradient Direction Histogram (PHOG), and Edge Direction Histogram Based on Local Binary Pattern operator (LBP-EOH). For illustration purpose, Deep Belief Network preparation was prepared by converting the original image to gray-scale image, normalizing to 40×40 pixels, then finally converting to binary image.

The second step was feature extraction. There were three features to be extracted in this step, i.e. Deep Belief Network (DBN), Pyramid Gradient Direction Histogram (PHOG), and Edge Direction Histogram Based on Local Binary Pattern operator (LBP-EOH). After the features were obtained,

Figure 2.5: Vehicle identification and classification flowchart



feature fusion was done using Karhunnen-Loeve transform. The Karhunnen-Loeve transform is an optimal orthogonal transformation under the mean square error. Lastly, the classification process was done using multiple kernel learning with support vector machine. Figure 2.5 shows the flow of the proposed algorithm.

The authors carried out the following experiment. Six kinds of vehicle were sampled. There were 150 samples for each kind of vehicle. The number of training samples were 30, 60, 80, 100, and 120 in random. The number of testing samples were 30. The training samples were used to construct models. The models were validated with testing samples. The experimental results were obtained. The authors found that

1. Increasing the number of training samples can increase the vehicle recognition rate

2. DBN feature cannot be used to express the overall characteristics of the vehicle

3. Traditional method outperforms the proposed method if the training samples is not large

4. The proposed method can reach 95% accuracy rate if the training samples is large enough

2.6 Decision Tree-based Machine Learning Algorithm for In-node Vehicle Classification

Ying et al. [10] proposed an in-node microprocessor-based vehicle classification approach. The authors used a 3-axis magnetometer sensor to detect and collect vehicles information. The information was then classified using a machine learning algorithm. Although, the authors did not focus on vehicle classification, a number of feature combination was the main purpose of the article.

In their experiment, a Honeywell HMC5883L magnetometer sensor was used to collect vehicles information. MSP430 microprocessor, from Texas Instrument, was chosen to be the main processor in their work. The input data for this system was the magnitude read from 3-axes magnetometer, i.e. x-axis, y-axis, and z-axis. The direction of each axis is described as follows:

- X-axis was parallel to the traveling route of the vehicle.
- Z-axis was pointing up out of the floor.
- Y-axis was not used because it could cause false positives depending on the ferrous make-up of the adjacent vehicle.

The authors obtained 12 features from magnetometer sensor, i.e. minimum, maximum, mean, and range of data from all 3 axes, for each vehicle. Note that, vehicles using in their experiment were radio controlled vehicles. In order to obtain the class of each vehicle, the collected data was categorized

Table 2.4: Initial classification rates with various features

Initial Cross-Validation Percentages	
Number of Features (Attributes)	Accuracy (%)
Two Features (minx, maxx)	94.00
Three Features (minx, maxx, maxz)	98.00
Four Features (miny, maxx, meanz, rangez)	98.86
Five Features	98.86

using a machine learning algorithm. The authors selected a Decision tree with C4.5 as its model building algorithm. They chose this model because of its simplicity for implementation and high computational efficiency.

Although, several features were obtained, only combinations between two to five were used. This was done to keep the final Decision tree models small. After the models were obtained, they were then validated using 10-fold cross-validation. Table 2.4 shows the best accuracy result of each number of combinations.

The result shows that classification rate could not be increased after the combination of four features. In addition, this result was obtained using radio controlled vehicles instead of real vehicles. Therefore, the result would be significantly different in real-world testing.

The authors concluded that their approach would be applicable in real-world scenario. This is because the only difference is the magnitude and length of the magnetic field generated from larger vehicles.

From the literature review, several approaches and techniques were investigated. Some of these were studied and used in this work. The detail of each technique used in this work will be explained in the following chapter.

Chapter 3

Methodology

Saripan et al. [8] had proposed a system which can solve the original problem concerning manually search as previously mentioned in chapter 1. However, the system's accuracy was not satisfied. Thus, there are possibilities to improve the system's accuracy, especially in feature extraction and classification module.

From the literature review, several techniques were discussed. These techniques can be applied to improve the accuracy of the feature extraction and the classification module. In case of feature extraction module improvement, most of the literature review proposed the features which are sophisticated to extract. Although, the researchers reported that using these features resulted in high accuracy, the implementation was more difficult and the user's usability was reduced. On the other hand, in order to improve the classification module, there were several methods proposed in the literature review. Most of the works applied ensemble methods, i.e. Boosting, Bagging, and Random forest. This improvement can be done without reducing user's usability since the classification module has no user's interaction. Moreover, applying on this dataset is expected to improve the accuracy of the classification module. The detail explanation of the selected approaches will be discussed as in the following sections.

The system proposed in [8] consists of three modules, i.e. feature extraction, classification and search manager. The searching process begin with the feature extraction module. The module receives a video from a user. Vehicles' records are extracted from the video and sent to the classification module. The extracted vehicle's characteristics are shown in the following list.

- Bounding box
  - Coordinate of top-left corners
  - Width (pixels)
  - Height (pixels)
- Color
  - Hue
  - Saturation
  - Value
- Detected vehicle foreground ratio

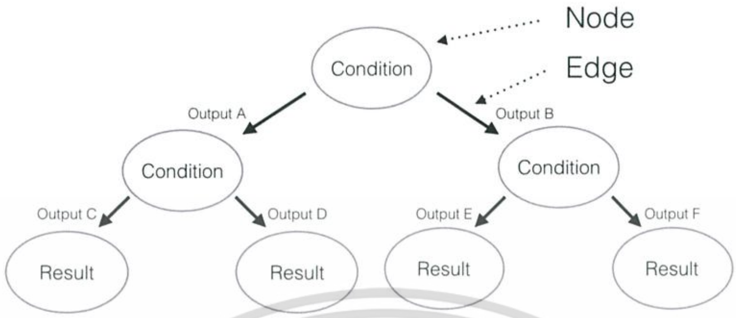
The extracted records are categorized into classes, i.e. type and color. There are three classes for type, i.e. Small, Medium, and Large, and six classes for color, i.e. Red, Yellow, Green, Blue, Black, and White. In addition, an Unknown class is added to both type and color classification. This class is added for the case that the classification result is not fit to any previously mentioned classes for both type and color. After the records are categorized, the records and its results are sent to the search manager module for result filtering.

There are two machine learning models in classification module, one for vehicle's type and the other is for vehicle's color. In the work of [8], these two models were based on Decision tree. In order to improve the classification module, several chosen machine learning algorithms are studied and explained in the following sections.

3.1 Machine learning algorithms

From literature review, it is found that the features extracted by the feature extraction module proposed by [8] are similar to other works. In contrast, most of the classification methods proposed in the literature review are not tested with the real vehicles' records. Therefore, the improvement

Figure 3.1: Example of a Decision tree model



of the classification module is the main focus of this work. There are four selected classification methods which will be tested, i.e. Decision tree (C4.5), Adaboost based on Decision tree (C4.5), Bagging based on Decision tree (C4.5), and Random forest. The detail of the algorithms is explained in the following sections.

3.1.1 Decision tree

Decision tree is a supervised machine learning algorithm. The model resulted from this algorithm has a tree-like structure. The model consists of two main components, i.e. nodes and edges. In general, there are two kinds of nodes, i.e. leaf node and non-leaf node. A leaf node contains the classification result while a non-leaf node carries a condition which is used to test the input data. The possible test results are shown in the second component, i.e. edges. Each edge connects two nodes and shows which node the input data has to be processed next. An example of Decision tree model is shown in Figure 3.1.

The classification process starts at the root node. The input data is tested with the condition inside that node. After the result is obtained, the process continues along the edge according to the result. The edge will lead the input data to the next testing node. The process iterates until there is no more edge to continue, that is the leaf node is reached. The classification result is determined by the class on the leaf node.

In order to obtain a Decision tree model, a model building algorithm is required. There are several Decision tree model building algorithms available, e.g. ID3, C4.5, etc. The idea of these algorithms is to split the training set into smaller subsets. In each spilt, it tries to make the subsets as homogeneous as possible. In order to determine the best split feature, information theory is used. Most of the Decision tree model building algorithm based on three approaches, i.e. Shannon entropy, Information gain, and Information gain ratio.

Shannon entropy determines uncertainty of the given data. If X is a random variable with possible values $\{x_1, x_2, ..., x_n\}$, the entropy can be written as follows:

$$Entropy(X) = -\sum_{i=1}^{n} p_i \times \log_2 p_i \tag{3.1}$$

where p_i is a probability of $X = x_i$.

Shannon entropy is then used to obtain the Information gain value. This obtained value determines the amount of information available from such data. For ID3 algorithm, the feature which has the highest Information gain is selected to be the test feature or test condition. Information gain can be defined using Equation (3.2) where P is probability of distribution and A is the selected attribute.

$$InfoGain(T, a) = Entropy(T) - \sum_{i=1}^{n} p_{a,i} \times Entropy(p_{a,i}) \tag{3.2}$$

where T is a set of training example and $p_{a,i}$ is the probability of event $a \in T$.

This approach works well on nominal value. However, for numerical value, Information gain does not give any useful information. Fortunately, it can be used to compute for Information gain ratio. This approach is known as C4.5. Information gain ratio is a normalization version of Information gain. The equations of Information gain ratio and Split information are shown as follows:

$$GainRatio(T, a) = \frac{InfoGain(T, a)}{SplitInfo(T, a)} \tag{3.3}$$

$$SplitInfo(T,a) = -\sum_{i=1}^{n} p_{a,i} \times \log_2 p_{a,i} \tag{3.4}$$

Based on Information Gain Ratio, which has been computed for each attribute, C4.5 selects the feature with the highest Information Gain Ratio to be the test feature. For detail explanation on Decision tree model and its construction process see [4, 5].

3.1.2 Ensemble methods

Ensemble methods are the method which combine several weak classifiers to make a new stronger classifier. The aims of the method is to obtain a more accurate classifier from decision scheme of weaker classifier. Examples of the ensemble methods from the literature review are Bagging, Boosting, and Random forest. These methods are used for object classification. Although, these classifiers share the same concept, they are different in subtle details. The explanation of Bagging, Boosting, and Random forest along with their differences are discussed as follows.

Bagging or Bootstrap aggregation is one of the ensemble methods which assigns the same priority to all classifiers. These classifiers are used to classify the same testing data set. However, the outputs will be different due to the different training data set which is used to train the classifiers. The based classifier can be any type. The results obtained from each based classifiers will be used to compute for statistical mean. In case of nominal value, majority votes can be applied. It should be noted that other selecting scheme can be used as well.

Boosting is another ensemble methods which is similar to Bagging. However, it has some differences in training process. In this method, the classifiers are trained in sequential aspect. That is the error from the previous classifier is also used in the training process of the next classifier. As a result, the classifiers in the late sequence tend to be dominant in giving the final output. Hence, each classifier has different priority which leads to different weight. The output is then calculated from a weighted sum of these classifiers.

While Bagging and Boosting can use any type of based classifier. The based classifier for Random forest is restricted to be a Decision tree. Furthermore, Random forest randomly picks the features to train for each based classifier. The rule of thumb for number of features is $\sqrt{N}$ where N is the number of available features. For the selecting scheme, Random forest can use the same approaches as in Bagging or Boosting.

In this work, Decision tree (C4.5), Boosting based on Decision tree (C4.5), Bagging based on Decision tree (C4.5), and Random forest will be applied to the dataset then compared. The most appropriate classifier will be selected to be the model for type and color classification.

3.2 Dataset preparation

In order to acquire the result of each machine learning algorithm, a dataset is required. A video shall be taken using a digital camera with a minimum specification as follows:

- Video's resolution must be at least 480×320 pixels
- Framerate must be at least 25 frames per seconds
- Weather condition must be
  - No rain
  - No cloud
  - No casting shadow
  - Daytime

The video shall be processed using the feature extraction module to obtain the vehicles' records. The records' attributes shall be divided into two groups, i.e. type and color. The details of each group is specified as follows:

- Type
  - Bounding box's position
  - Bounding box's size
  - Vehicle appearance area
  - Target size

- Color
  - Hue
  - Saturation
  - Luminance
  - Target color

Note that, the target classes of each record shall be determined manually.

3.3 Performance evaluation

For validation process, the result of each classification method shall be compared in term of training time and true positive rate. The true positive rate percentage of each method is determined using the number of correctly classify records divided by the total number of records time 100, as shown in Equation 3.5.

$$TruePositiveRate = \frac{CorrectRecords}{TotalRecords} \times 100\% \tag{3.5}$$

In addition, a confusion matrix of each classification result shall be used for detail analysis. This analysis results is then used for further classification improvement.

The details of each proposed method, data preparation, and performance evaluation are discussed in this chapter. Next chapter will show experiment in applying these proposed methods on the collected dataset. Moreover, the results and performance evaluation will be discussed.

Chapter 4

Experimental Setup and Results

The system proposed by Saripan et al. [8] can solve the problems explained previously. However, such solution requires further improvement especially in the overall system's accuracy. Chapter 3 discussed on how to improve the classification module and feasible approaches. This chapter explains the experimental setup and results for each method discussed in chapter 3.

According to the discussion in the previous chapter, the classification module is the main focuses of this work. The classification module has to categorize vehicle into two types of classes. These two target classes are type and color. For each target class, a dedicate classifier is required to label each vehicle record.

For type classification, the vehicles are classified into four classes, i.e. Small, Medium, Large, and Unknown while vehicles' color are classified into seven classes, i.e. Black, White, Red, Blue, Yellow, Green and Unknown. Note that, Unknown is placed in both classification result in case that the model cannot classify the input data. The result of this class requires further inspection from an expert.

The input for this module is expected to be provided by the feature extraction module. The data processed by this module can be split into three

Table 4.1: Detail information of the dataset video

Duration	763	seconds
Width	640	pixels
Height	480	pixels
Frame rate	25	frame per second

groups, i.e. bounding box, color, and vehicle appearance area. The bounding box refers to a rectangle which covers the input vehicle in the video frame. Next feature, vehicle's color, is given in HSL color space. While the last feature, vehicle appearance area, is the ratio of number of pixels between vehicle and background. The following list shows the input data attributes in detail.

- Bounding box
  - X coordinate of top-left corner
  - Y coordinate of top-left corner
  - Width
  - Height
- Color
  - Hue
  - Saturation
  - Luminance
- Vehicle appearance area

In chapter 3, Decision tree, Bagging, Boosting, and Random forest were proposed as feasible methods for the classification module. In order to train and test these models, a dataset for training and testing is needed to be prepared. A digital camera is used to take a video to produce the dataset. The video is prepared under the following conditions, i.e. daytime, no cloud, no rain, and no casting shadow. The video's duration is about 13 minutes. The detail information of the video is shown in Table 4.1.

Table 4.2: Training data matrix

	Black	White	Red	Blue	Yellow	Green	Unknown	Total
Small	51	127	15	9	6	6	21	235
Medium	69	185	4	2	4	4	42	310
Large	0	72	0	0	0	0	5	77
Unknown	1	9	0	0	0	0	54	64
Total	121	393	19	11	10	10	122	686

Table 4.3: Testing data matrix

	Black	White	Red	Blue	Yellow	Green	Unknown	Total
Small	15	25	0	0	6	0	4	50
Medium	12	57	2	3	0	2	21	97
Large	0	29	0	0	0	0	2	31
Unknown	0	11	0	0	0	0	39	50
Total	27	122	2	3	6	6	66	228

In order to obtain the dataset, the video is processed using the feature extraction module on a desktop computer with the following specifications: CPU Intel Core i5-3470, RAM DDR3 8GB 1600. The module gives 914 records of detected vehicle with their characteristics. The records are randomly picked to split the dataset into training set and testing set, 686(75%) and 228(25%), respectively. The detail of training records are in training set and testing set are shown in Table 4.2 and Table 4.3.

For type classifier, bounding box and vehicle appearance are used to produce the model while color in HSL color space is used to build the color classifier. The training set is used to train and test each model. The performance of each classifier based on training dataset is shown in Table 4.4 - 4.11.

The results of type classification show that most classifiers made mistake in Small-Medium classification more than others. This is because the dimensions of Small and Medium vehicle are more difficult to identify compared to Medium and Large.

Table 4.4: Confusion matrix of size classifier based on Decision tree for training dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	203	32	0	0
	Medium	31	278	0	1
	Large	0	1	71	5
	Unknown	2	3	1	58

For color classification, the most incorrect classification result is between White and Unknown, including Silver, Gray, etc. This can be explained by observing the images shown in Figure 4.1. It is obvious that the color of both vehicles are similar and difficult to classify even using human inspection.

Figure 4.1: Example of misclassify vehicle color

(a) White vehicle (b) Silver vehicle

In addition, Table 4.12 shows training time and accuracy of each classifier on the training set. The result shows that Decision tree can be trained faster than other methods but it has the lowest accuracy. On the other hand, Random forest manages to obtain nearly 100% accuracy rate in trading of slower training time. It is found that Decision tree uses ten times faster compared to Random forests. Although, Random forests can outperforms other methods, it also signified the overfitting problem.

Despite the fact that Random forests might be the best classifier to be used in the classification module. It cannot be confirmed by using training

Table 4.5: Confusion matrix of size classifier based on adaboost(Decision tree) for training dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	216	19	0	0
	Medium	17	292	0	1
	Large	0	1	75	1
	Unknown	3	2	1	58

Table 4.6: Confusion matrix of size classifier based on Bagging(Decision tree) for training dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	213	22	0	0
	Medium	23	283	0	4
	Large	0	4	69	4
	Unknown	3	4	2	55

Table 4.7: Confusion matrix of size classifier based on Random forest for training dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	235	0	0	0
	Medium	0	310	0	0
	Large	0	0	77	0
	Unknown	0	0	0	64

Table 4.8: Confusion matrix of color classifier based on Decision tree for training dataset

		Classified as						
		Red	Yellow	Green	Blue	Black	White	Unknown
Actual	Red	11	2	0	1	4	0	1
	Yellow	0	8	0	0	2	0	0
	Green	0	0	0	0	8	2	0
	Blue	5	0	0	6	0	0	0
	Black	0	0	0	0	103	10	8
	White	0	0	0	0	14	368	11
	Unknown	0	0	0	0	15	63	44

Table 4.9: Confusion matrix of color classifier based on adaboost for training dataset

		Classified as						
		Red	Yellow	Green	Blue	Black	White	Unknown
Actual	Red	14	0	0	0	4	0	1
	Yellow	0	8	0	0	2	0	0
	Green	0	0	5	0	3	2	0
	Blue	0	0	0	9	2	0	0
	Black	0	0	0	0	109	11	1
	White	0	0	1	0	11	376	5
	Unknown	0	0	2	0	14	64	42

Table 4.10: Confusion matrix of color classifier based on Bagging for training dataset

		Classified as						
		Red	Yellow	Green	Blue	Black	White	Unknown
Actual	Red	12	0	0	2	4	0	1
	Yellow	0	10	0	0	0	0	0
	Green	0	0	4	0	3	1	2
	Blue	0	0	0	10	1	0	0
	Black	0	0	1	0	109	5	6
	White	0	0	0	0	10	376	7
	Unknown	0	1	0	0	4	46	71

Table 4.11: Confusion matrix of color classifier based on Random forest for training dataset

		Classified as						
		Red	Yellow	Green	Blue	Black	White	Unknown
Actual	Red	18	0	0	0	0	1	0
	Yellow	0	10	0	0	0	0	0
	Green	0	0	10	0	0	0	0
	Blue	0	0	0	11	0	0	0
	Black	0	0	0	0	121	0	0
	White	0	0	0	0	0	392	1
	Unknown	0	0	0	0	0	0	122

Table 4.12: Type and color classifier train results

Method	Training time(s)		Accuracy(%)	
	Type	Color	Type	Color
Decision tree	**<0.01**	**0.01**	88.92	78.71
Adaboost	0.04	0.08	93.44	82.07
Bagging	0.18	0.06	90.37	86.29
Random forest	0.10	0.11	**100.00**	**99.70**

Table 4.13: Confusion matrix of size classifier based on Decision tree for testing dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	39	11	0	0
	Medium	4	91	1	1
	Large	0	0	13	18
	Unknown	7	3	2	38

Table 4.14: Confusion matrix of size classifier based on Adaboost(Decision tree) for testing dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	38	12	0	0
	Medium	8	87	1	1
	Large	0	0	14	17
	Unknown	7	3	2	38

dataset. Therefore, the testing dataset is applied to the models for real-world scenario simulation. This scenario refers to the input vehicle records for the classification module which are not seen before. Table 4.13 - 4.20 show the test results of each classifier using the testing dataset.

It can be seen that the highest accuracy for type and color classification are obtained from Random forest and Adaboost, respectively. However, the results also show that all the methods are comparable in term of accuracy.

In order to obtain the overall classification accuracy, the accuracy of type and color classifiers are combined. Both classifiers must give correct classification for such data in order to obtain overall correct classification. From the individual test result, Decision tree and Random forest are the top two classifiers for type classification while the top two classifiers for color are Adaboost and Bagging. Hence, the combination of these classifiers are expected to give high accurate results.

Table 4.15: Confusion matrix of size classifier based on Bagging(Decision tree) for testing dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	40	10	0	0
	Medium	3	88	0	6
	Large	0	0	13	18
	Unknown	8	2	5	35

Table 4.16: Confusion matrix of size classifier based on Random forest for testing dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	41	9	0	0
	Medium	4	91	0	2
	Large	0	0	15	16
	Unknown	7	3	5	35

Table 4.17: Confusion matrix of color classifier based on Decision tree for testing dataset

		Classified as						
		Red	Yellow	Green	Blue	Black	White	Unknown
Actual	Red	0	0	0	0	2	0	0
	Yellow	0	4	0	0	2	0	0
	Green	0	0	0	0	2	0	0
	Blue	1	0	0	1	1	0	0
	Black	0	0	0	2	20	2	3
	White	0	0	0	0	1	115	6
	Unknown	0	0	0	0	12	39	15

Table 4.18: Confusion matrix of color classifier based on adaboost(Decision tree) for testing dataset

		Classified as						
		Red	Yellow	Green	Blue	Black	White	Unknown
Actual	Red	0	0	0	0	2	0	0
	Yellow	0	4	1	0	1	0	0
	Green	0	0	0	0	2	0	0
	Blue	0	0	0	2	1	0	0
	Black	0	0	0	1	24	2	0
	White	0	0	0	0	1	115	6
	Unknown	0	0	1	0	11	42	12

Table 4.19: Confusion matrix of color classifier based on Bagging(Decision tree) for testing dataset

		Classified as						
		Red	Yellow	Green	Blue	Black	White	Unknown
Actual	Red	0	0	0	0	2	0	0
	Yellow	0	4	1	0	1	0	0
	Green	0	0	1	0	1	0	0
	Blue	1	0	0	0	2	0	0
	Black	0	0	0	1	22	2	2
	White	0	0	0	0	1	109	12
	Unknown	0	0	0	0	6	38	22

Table 4.20: Confusion matrix of color classifier based on Random forest for testing dataset

		Classified as						
		Red	Yellow	Green	Blue	Black	White	Unknown
Actual	Red	0	0	0	0	2	0	0
	Yellow	0	4	2	0	0	0	0
	Green	0	0	1	0	1	0	0
	Blue	0	0	0	1	2	0	0
	Black	0	0	0	2	20	2	3
	White	0	0	0	0	2	104	16
	Unknown	1	0	0	0	5	34	26

Table 4.21: Type and color classifier test results

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

Table 4.22: Result of combined classifiers based on testing dataset

Combined Method		Accuracy(%)
Type	Color	
Decision tree	Decision tree	55.26
Decision tree	**Adaboost**	**56.57**
Decision tree	**Bagging**	**57.89**
Decision tree	Random forest	55.26
Adaboost	Decision tree	53.07
Adaboost	Adaboost	53.94
Adaboost	Bagging	54.38
Adaboost	Random forest	52.19
Bagging	Decision tree	54.82
Bagging	Adaboost	55.70
Bagging	Bagging	56.14
Bagging	Random forest	53.94
Random forest	Decision tree	55.70
Random forest	**Adaboost**	**56.57**
Random forest	**Bagging**	**57.01**
Random forest	Random forest	54.82

The combined classifiers test results are shown in Table 4.22. Combination of Decision tree and Bagging gives the highest accuracy rate. However, the results do not show significant difference from the other combinations.

In consideration of computation time, Decision tree is the algorithm which gives the fastest training time for both type and color classifiers. However, it is found that Bagging surpasses most of the methods for color classification except Decision tree. Considering the overall performance, the combination of Decision tree and Bagging gives better accuracy rate while maintain low training time. Therefore, for this work, combination of Decision tree and Bagging is chosen to be the classifiers for type and color, respectively.

Chapter 5

Conclusion

This research improves the classification module implemented in [8]. The improvement concerning accuracy and computation time. There are two possible approaches, i.e. feature modification and classifier modification. This work focuses on classifier modification using Ensemble methods.

There are several Ensemble methods introduced in this work, e.g. Adaboost, Bagging, Random forest. These methods are applied to both vehicle's type and color classification. It is found that, for type classification, Random forest gives the highest accuracy of 79.82% with training time 0.10 seconds. However, the accuracy of Decision tree is found to be 79.38% with much more lower training time of less than 0.01 seconds. For this reason, this work selects Decision tree to be the classifier for vehicle's type.

For color classification, Adaboost gives the most accurate result of 69.29%. This result is not significantly different from the other classifiers, e.g. Bagging with accuracy of 68.85%. However, in consideration of training time, Bagging can be trained faster for 33.33%. Therefore, Bagging algorithm is selected to be color classifier for classification module.

This work also concern the overall accuracy which combines the accuracy of both type and color classifiers. The result shows that the combination of Decision tree and Bagging is suitable to be the type and color classifier for the classification module. This combination gives accuracy of 57.89% with training time less than 0.07 seconds.

As future work, the system can be improved further. For instance, new features can be introduced to the feature extraction module. This might also benefit the classification module in term of accuracy due to more variation of input data. For feature extraction module, computation time can be reduced by using parallel programming technique. In addition, for classification module, alternative machine learning algorithms can be introduced, e.g. neural network, deep learning, extreme learning machine, etc.

References

[1] S. B. Changalasetty, A. S. Badawy, L. S. Thota, and W. Ghribi. Moving vehicles classification in weka. In *2015 International Conference on Advanced Computing and Communication Systems*, pages 1–6, Jan 2015.

[2] R. Feris, B. Siddiquie, Y. Zhai, J. Petterson, L. Brown, and S. Pankanti. Attribute-based vehicle search in crowded surveillance videos. *ICMR'11*, apr 2011.

[3] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson. Random forests for land cover classification. *Pattern Recognition Letters*, 27(4):294 – 300, 2006. Pattern Recognition in Remote Sensing (PRRS 2004).

[4] P. Harrington. *Machine learning in action.* Manning Publications Co., 2012.

[5] B. Hssina, A. Merbouha, H. Ezzikouri, and M. Erritali. A comparative study of decision tree ID3 and C4.5. *Special Issue on Advances in Vehicular Ad Hoc Networking and Applications*, pages 13–19, 2014.

[6] R. Lienhart, L. Liang, and A. Kuranov. A detector tree of boosted classifiers for real-time object detection and tracking. In *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference On*, volume 2, pages II–277–80 vol.2, 2003.

[7] H. Qian, Y. Zhang, and C. Liu. Vehicle classification based on the fusion of deep network features and traditional features. In *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)*, pages 257–262, March 2015.

[8] K. Saripan, C. Deachakul, and W. Maungmai. *Smart Vehicle Search in Surveillance video.* King Mongkut's Institute of Technology Ladkrabang, 2015.

[9] B. Wu and R. Nevatia. Cluster boosted tree classifier for multi-view, multi-pose object detection. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[10] K. Ying, A. Ameri, A. Trivedi, D. Ravindra, D. Patel, and M. Mozumdar. Decision tree-based machine learning algorithm for in-node vehicle classification. In *2015 IEEE Green Energy and Systems Conference (IGESC)*, pages 71–76, Nov 2015.

Honorary Chairs
Monai Krairiksh, KMITL
Prabhas Chongsatitwattana, CU

International Advisory Committee
Prayoot Akkaraekthalin, KMUTNB
Somsak Choomchuay, KMITL
Kosin Chamnongthai, KMUTT
Yasumasa Fujisaki, SICE, Japan
Dong-il Dan Cho, ICROS, Korea
Yoshihiro Matsui, TNCT, Japan
David Banjerdpongchai, CU
Tomoaki Sato, Hirosaki U, Japan
Pornchai Supnithi, KMITL

General Chair
Somsak Choomchuay, KMITL

General Co-chair
Kosin Chamnongthai, KMUTT

Technical Program Chairs
Chuwong Phongcharoenpanich, KMITL
Krischonme Bhumkittipich, RMUTT
Eryk Dutkiewicz, UTS, AU

Special Session Chairs
Sathaporn Promwong. KMITL
Roungsan Chaisricharoen, MFU

Tutorial Chair
Keattisak Sripimanwat

Publication Chairs
Uthen Kamnarn, RMUTL
Apirath Limmanee, BUU

Publicity Chair
Wannaree Wongtrairat, RMUTI
Petcharat Suriyachai, PSU
Soohee Han, POSTECH, Korea
Kou Yamada, SICE, Japan

Registration Chair
Panuwat Janpugdee, CU

Finance Chairs
Wiboon Prompanich, KMITL
Pairin Kaewkuay, ECTI

Local Arrangement Chairs
Krisda Yingkayan, RMUTL
Pichaya Tandayya, PSU

Information System Chair
Akkarat Boonpoonga, KMUTNB
Watid Phakphisut, KMITL

Secretariat
Theerayod Wiangtong, MUT

Organized by
ECTI Association

Technical Co-sponsor
IEEE Thailand Section
SICE, Japan
ICROS, Korea
REV, Vietnam

The 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology or ECTI-CON 2017 (IEEE Conference Record Number #38636) is the fourteenth annual international conference organized by Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association, Thailand. The conference aims to provide an international platform to present technological advances, launch new ideas and showcase research work in the field of electrical engineering, electronics, computer, telecommunications and information technology. Accepted papers will be published in the Proceedings of ECTI-CON 2017 and will be submitted for inclusion in the IEEE Xplore. Acceptance will be based on quality, relevance and originality.

Technical Tracks

1. Devices, Circuits and Systems
2. Computers
3. Information Technology
4. Communication Systems
5. Controls, Instrumentation and Measurements
6. Electrical Power Systems
7. Power Electronics
8. Signal Processing
9. Other Related Areas
10. Special Sessions

Special Sessions

A proposal for a special session can be submitted to the special session chair before the deadlines. The session topic can be varied upon one's interest but still relate to the role of Electrical / Electronic Engineering, Computer, Telecommunication, Computer and IT.

Best Paper Awards

Paper with the highest score of a track that holds more than 10 papers will be nominated as a "Best Paper Award" paper.

Important Dates

Deadline for Special Session Proposal	15 February 2017
Deadline for Submission	**15 March 2017**
Notification of Acceptance	30 April 2017
Deadline for Final Manuscript Submission	15 May 2017
Deadline for Early Registration	15 May 2017
Conference Dates	27-30 June 2017

Paper Submissions

1) Prospective authors are invited to submit original full papers WITHOUT authors' names and affiliations, in English, of 2-4 pages in standard IEEE two-column format only, reporting their original work and results, applications, and/or implementation in one or more of the listed areas.
2) Papers must be submitted online only through the submission system of the conference website.
3) At least one author of each accepted paper MUST register and present the paper at the conference in order that the paper is to be included in the program. The program will also be submitted for inclusion in the IEEE Xplore.

Further Publication

Potential papers are encouraged for their extension and submit to ECTI Journals (ECTI-EEC or ECTI-CIT) for further publication.

More information is available at http://www.ecti-con.org/con-2017/
https://www.facebook.com/ECTICON2017/

2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)

Tree-Based Vehicle Classification System

Kiatkachorn Saripan
International College,
King Mongkut's Institute of Technology Ladkrabang,
Bangkok, Thailand
Email: 55090007@kmitl.ac.th

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

***Abstract*—Nowadays, traffic surveillance systems are installed in major cities. They are usually used for two purposes, i.e. real-time traffic monitoring and archived events searching. For the latter purpose, the traffic surveillance systems can be used for police officers' benefits, such as vehicle identification in specific events including stolen vehicles or hit-and-run cases. In such circumstances, the officers are required to identify the vehicle in archived videos according to its appearances. This task is usually accomplished manually through visual perception. The problems arise from this approach Even though this approach results in good accuracy, it is time consuming and prone to error due to human fatigue for long duration videos. In order to solve these problems, a tree based vehicle classification system is proposed. This system consists of three modules, i.e. feature extraction, classification, and search manager. The feature extraction module is used for image and video processing. It extracts the desired features to be used further in the classification module. The classification module uses these features and results in pre-defined vehicle classes. The classification results are stored in the search manager module for further filtering according to user's query command. This paper focuses on the classification module. There are two features designed to be used in the proposed classification module, i.e. types and colors. Vehicles are classified into four classes of type and seven classes of color. Several tree based algorithms are applied to the dataset. The experimental results show that all the algorithms are comparable. However, the highest accuracy for type and color classification are obtained by using decision tree and bagged decision tree, respectively.**

I. Introduction

In major cities, there are several surveillance cameras installed to capture traffic conditions and events which can occur unpredictably. In general, surveillance video can be used for two main purposes, i.e. traffic monitoring and events searching. In this paper, the authors focus only on events searching.

This research categorizes events into two groups, i.e. major incidents and minor incidents. The major incidents referred to incidents which are reported on the news. These incidents are usually easy to identify the time of occurrence. Hence, the incidents of interest can be found more quickly. On the other hand, the minor incidents are usually not reported. This means that there is less information concerning time of occurrence and hence more difficult to be found. These minor incidents include stolen vehicle, minor accidents, etc.

In order to investigate these minor incidents, police officers need to spend a lot of time searching manually through the surveillance videos. For example, in vehicle searching officers receives vehicle's characteristic, e.g. vehicle's color and type, approximate missing time, etc., and are required to search for a time of vehicle's appearances in videos. Moreover, after a period of searching, the officers might have tiredness which leads to missing some important evidences. This results in low searching accuracy.

In order to decrease searching time and increase searching accuracy, a method of vehicle classification can be applied for such tasks. Currently, there are many proposed methods which can be applied to these problems. P. O. Gislason et al. [1] compared several classification methods, i.e. classification and regression tree, bagging, boosting, and random forest. Their experimental result showed that all of the classification methods were comparable to one another. However, they chose random forest for their work for many reasons: shorter learning time compared to other ensemble methods, required no guidance, etc. S. B. Changalasetty et al. [2] proposed a vehicle classification system which classified moving vehicles into two categories, i.e. big and small. The authors proposed artificial neural network as a classifier. They reported that it achieved more than 90% accuracy. K. Ying et al. [3] used decision tree as a classifier for their vehicle classification. In their experiment, they tried various number of feature combinations in order to keep the final decision tree small. This led to using less memory and less computational time. However, they couldn't increase classification accuracy in case of four or more combined features were applied. The authors performed experiment on radio controlled car models and they claimed that this approach would be applicable to the real scenario. W. Ma et al. [4] presented a method based on multi-feature fusion. This method combined two features, local and global, together. The local feature was acquired by using Scale-Invariant Feature Transform (SIFT) while global feature is acquired by using Principle Component Analysis (PCA). The authors used multi kernel learning to classify vehicles into three classes, i.e. car, bus, and truck. They reported that the accuracy of car, bus, and truck classes were 93.3%, 88.3%, and 87.0%, respectively. R. Feris et al. [5] presented an application which could search for vehicles in surveillance videos. They proposed a new classifier call motionlet classifier. This motionlet classifier is a detector based on Adaboost learning [6]. In their experiment, one motionlet detector could detect vehicle only in one direction. They used

978-1-5386-0449-6/17/$31.00 ©2017 IEEE

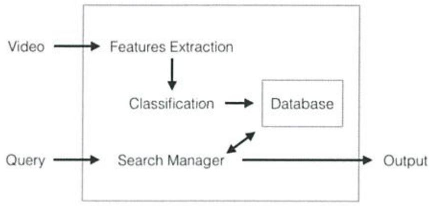


Fig. 1: System overview

twelve motionlet detectors in their projects to detect vehicle which may come in different angle. As a result, their classifier achieved 87% accuracy rate. B. Hssina et al. [7] explained in detail about decision tree and its model building algorithms, e.g. ID3, C4.5. In addition, they also compared the decision tree building algorithms in term of accuracy and execution time. They had concluded that C4.5 gave more accuracy and executed faster compared to ID3.

This paper proposes a system to circumvent vehicle searching in surveillance videos by using tree-based vehicle classification system. The proposed system aims to search for objects in the given video by using some specific vehicle's characteristics. The searching duration is expected to be less than the video's duration. Furthermore, by using this approach the human's fatigue problem is eliminated. The proposed system consists of three main modules, i.e. feature extraction, classification, and search manager. Fig. 1 shows the system overview. The feature extraction module is designed to process the input video by using image processing techniques. The extracted information is sent to the classification module to categorize vehicles into classes. Finally, the search manager module stores and filters the results according to the given query command. This work focuses mainly on the classification module which used tree-based classification method.

This paper is organized as follows. Section II explains the proposed methods in detail. The experimental setup of the proposed methods is discusses in section III. The accuracy of the result and its comparison are also given. Section IV concludes the experimental results and discusses about future works.

II. Proposed Method

A tree-based vehicle classification system requires two inputs, i.e. a surveillance video and vehicle's characteristics. The system consists of three modules, i.e. feature extraction, classification, and search manager. In the beginning, the video is fed into the feature extraction module. The module processes the video in order to extract the features which are shown in the Table I.

The extracted data is then sent to the classification module. There are two classifiers, type and color, implemented in the classification module. The type classifier categorizes vehicle into four possible classes, i.e. small, medium, large, and unknown. The color classifier classifies vehicle into seven possible classes, i.e. black, white, blue, green, yellow, red, and

TABLE I: Extracted features from the feature extraction module

Bounding box	Color	Ratio
- Coordinate of top-left corner	- Hue	- Detected vehicle
- Width	- Saturation	foreground ratio
- Height	- Luminance	

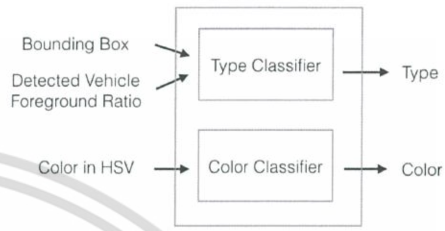


Fig. 2: Classification module overview

unknown. Note that, both classifiers contain an unknown class. This class contains the vehicle with ambiguous characteristics. The results from both classifiers are sent to the search manager module. This module then stores and filters the results according to the given query command. Fig. 2 shows the overview of the classification module.

According to [1], the authors reported that a tree-based classifier provided good accuracy. For this reason, such type of classifier is chosen to implement in this work. The implemented tree-based classifiers are decision tree (J48), bagging, boosting, and random forest. The detail of each implemented tree-based algorithm is explained as follows:

A. *Decision tree*

Decision tree is a supervised machine learning model. It has nodes and branches as similar to tree-based stucture model. In addition, it uses divide and conquer technique to categorize input data into classes. Nodes can be categorized into two types, leaf and non-leaf node. For non-leaf node, the input instance is tested based on the condition on the node. The result shows which branch does it have to test next. The process is iterated until the leaf node is reached. The leaf node then shows the target class for the input instance. In order to build a decision tree model, this work applies C4.5 algorithm [8]. The algorithm used formulas based on information theory to test the information gain on each feature. The feature which has the highest information gain is selected. Then, this feature is removed from the dataset. The test of information gain on features is iterated until homogeneity is reached or there is no more features left.

B. *Ensemble methods*

Ensemble methods aim to build a stronger classifier based on weak classiers. For example, bagging uses many classifiers to classify the same data. Then, computes for statistical mean for the final result. In case of nominal value, voting system is

TABLE II: Information of the dataset video

Duration	763 second
Width	640 pixels
Height	480 pixels
Frame rate	25 fps

used to give the final result. Boosting also uses many classifiers to classfy the same data. However, each classier has its own priority. The result from each classifer times by its priority and computes for statistical mean for final result. Random forest uses only decision tree as its classfiiers. For each decision tree in the forest, only subset of features are used to build the model. The final result are obtained the same way as bagging.

For this research, the classifiers will be evaluated in term of accuracy and execution time.

III. Experimental result

The experiment is conducted on a desktop computer with the following specifications: CPU Intel Core i5-3470, RAM DDR3 8GB 1600. The video is taken during daytime without rain, cloud, and casting shadow. The characteristics of video information is shown in Table II.

The feature extraction module is used to process the video to obtain vehicles' information. There are 914 records of vehicle's information extracted from the video. The records are split into two groups, i.e. training and testing. The percentage of the training and testing dataset are 75% and 25%, respectively. Training dataset is used to develop classifiers for both type and color. There are three features extracted from the video, i.e. bounding box, detected vehicle foreground ratio, and color. The first two are selected to be the features of type classifier while the last feature is used for color classifer.

Table III and IV illustrate confusion matrices based on decision tree for training dataset. It is found from Table III that most of the vehicle's types are correctly classified. However, some misclassifications occur especially between small and medium classes. This is because the size of these two classes are close to each other. Table IV shows that most colors can be classified correctly. However, it is found that many misclassifications occur between white, black and unknown. This is because silver and dark grey are included in unknown class. Hence, it is highly possible to be misclassified with white and black color. Table V and Table VI shows similar results to Table III and Table IV, respectively. However, the illustrated results are performed based on testing dataset. The confusion matrix information is also obtained for other methods including adaboost, bagging, and random forest.

The training time and training accuracy for each method are summarized in Table VII while the testing accuracy is illustrated in Table VIII. The results from Table VII show that decision tree can be trained faster than other methods but its accuracy is the lowest. On the other hand, random forest requires much more training time but it obtains nearly 100% accuracy. However, this might lead to an overfitting problem. Table VIII shows the accuracy for each classifier

TABLE III: Confusion matrix of type classifier based on decision tree for training dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	203	32	0	0
	Medium	31	278	0	1
	Large	0	1	71	5
	Unknown	2	3	1	58

TABLE IV: Confusion matrix of color classifier based on decision tree for training dataset

		Classified as						
		R	Y	G	BL	BK	W	U
Actual	R	11	2	0	1	4	0	1
	Y	0	8	0	0	2	0	0
	G	0	0	0	0	8	2	0
	BL	5	0	0	6	0	0	0
	BK	0	0	0	0	103	10	8
	W	0	0	0	0	14	368	11
	U	0	0	0	0	15	63	44

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

TABLE V: Confusion matrix of type classifier based on decision tree for testing dataset

		Classified as			
		Small	Medium	Large	Unknown
Actual	Small	39	11	0	0
	Medium	4	91	1	1
	Large	0	0	13	18
	Unknown	7	3	2	38

based on the testing dataset. It can be found that the highest accuracy for type and color classification is obtained from random forest and adaboost, respectively. However, in general, these accuracies are comparable to other classifiers.

In order to obtain the overall correct classification results, both type and color classifiers must give the correct result for such vehicle's information. From the individual testing result, decision tree and random forest are the top two classifiers for type classification while the top two classifiers for color classification are adaboost and bagging. The combination of these classifiers is expected to produce better accuracy result compared to others. Table IX shows the accuracy of the combined classifiers based on testing dataset. The highest classification rate is obtained by combining decision tree for type classification and bagging for color classification. In addition, in term of training time, bagging can be trained much faster than most of the methods. Therefore, in this work, decision tree and bagging are selected to be classifiers for the classification module. Examples of classification result obtained by applying the selected combined classifier are shown in Fig. 3.

(a) Small & White (b) Small & Black (c) Medium & Green (d) Medium & Red (e) Large & White

Fig. 3: Example of the final result given by the system

TABLE VI: Confusion matrix of color classifier based on decision tree for testing dataset

		Classified as						
		R	Y	G	BL	BK	W	U
Actual	R	0	0	0	0	2	0	0
	Y	0	4	0	0	2	0	0
	G	0	0	0	0	2	0	0
	BL	1	0	0	1	1	0	0
	BK	0	0	0	2	20	2	3
	W	0	0	0	0	1	115	6
	U	0	0	0	0	12	39	15

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

TABLE VII: Type and color classifier train results

Method	Training time(s)		Accuracy(%)	
	Type	Color	Type	Color
Decision tree	**<0.01**	**0.01**	88.92	78.71
Adaboost	0.04	0.08	93.44	82.07
Bagging	0.18	0.06	90.37	86.29
Random forest	0.10	0.11	**100.00**	**99.70**

TABLE VIII: Type and color classifier test results

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

IV. Conclusion and Discussion

This work proposes tree-based type and color classifiers for classification module in vehicle classification system. The results show that combination of decision tree and bagging gives higher accuracy and shorter training time compared to other methods performed in this work. The selected combined classifiers achieve 57.89% accuracy mainly due to misclassification from color classifier. For this reason, the authors aim to improve classification accuracy using two approaches. For vehicle's type classification, new features shall be investigated and added to the classifier while for color classification other algorithms, such as neural network, shall be experimented.

TABLE IX: Results of combined classifiers based on testing dataset

Combined Method		Accuracy(%)
Type	Color	
Decision tree	Decision tree	55.26
Decision tree	**Adaboost**	**56.57**
Decision tree	**Bagging**	**57.89**
Decision tree	Random forest	55.26
Adaboost	Decision tree	53.07
Adaboost	Adaboost	53.94
Adaboost	Bagging	54.38
Adaboost	Random forest	52.19
Bagging	Decision tree	54.82
Bagging	Adaboost	55.70
Bagging	Bagging	56.14
Bagging	Random forest	53.94
Random forest	Decision tree	55.70
Random forest	**Adaboost**	**56.57**
Random forest	**Bagging**	**57.01**
Random forest	Random forest	54.82

References

[1] 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.

[2] 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

[3] 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

[4] W. Ma, Z. Miao and Q. Zhang, "Vehicle classification based on multi-feature fusion," *5th IET International Conference on Wireless, Mobile and Multimedia Networks (ICWMMN 2013)*, Beijing, 2013, pp. 215-219. DOI: 10.1049/cp.2013.2411

[5] 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 Retrieval (ICMR '11)*. New York, 2011.

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

[7] B. Hssina, A. Merbouha, H. Ezzikouri and M. Erritali. "A comparative study of decision tree ID3 and C4.5," *International Journal of Advanced Computer Science and Applications(IJACSA)*, 2014.

[8] J.R. Quinlan. (1993). C4.5: Programs for machine learning. San Mateo, CA: Morgan Kaufmann In.

Author Biography

Author: Kiatkachorn Saripan

Degree: Master of Engineering

Date of Graduation: July 6^{th} 2017

Date of Birth: June 10^{th} 2017

Place of Birth: Bangkok, Thailand

Undergraduate and Graduate Education:
Master of Engineering in Computing in Engineering Systems,
King Mongkuts Institute of Technology Ladkrabang, Bangkok, 2016

Bachelor of Engineering in Software Engineering,
King Mongkuts Institute of Technology Ladkrabang, Bangkok, 2015

Major: Computing in Engineering Systems

Presentations and Publications:
[1] 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, Thailand, 2017, pp. 439-442.