CHAPTER 1

INTRODUCTION

1.1 Overview of Vehicle Detection and Classification System

A vehicle detection and classification system automatically detects and classifies vehicles from traffic surveillance video sequences. The system first detects moving vehicles on the road from a given video sequences. Then, the system classifies the detected vehicles into their respective categories. Vehicles are classified into car, lorry, motorcycle or non vehicles. Lastly, the system counts the number of vehicles according to their classes.

Vehicle detection and classification system can be used in the following applications:

1. Transportation planning for traffic operation and pavement design (Avely et al., 2004). Differences in size, weight, and performance between light and heavy vehicle need to be considered during transportation planning.

2. Vehicles' data collection for safety evaluation purposes. Studies show high percentages of fatal accidents involved heavy vehicles (Avely et al., 2004).

3. Road maintenance planning. Road wear is affected by weight of the vehicles and traffic flows (Hjort et al., 2008). Road with high traffic of heavy vehicles need to be maintained more frequently.

4. Traffic management. Traffic congestion can be managed by allowing only light vehicles to travel on a particular road during peak hours.

1

1.2 The Significant of the Project

Over the years, demands for processed vehicles information had increased. Processed vehicle information is very useful in many fields. These information can be obtained by a human agent or by a special intelligent system that can replace the human agent. The traditional way has proven to be very labor intensive and not reliable. As a result, a lot of detection and classification techniques have been developed. Examples of vehicle detection techniques include Gaussian Mixture Model (GMM) which is often used in extracting moving objects (Stauffer and Grimson, 1999), visual background extractor (ViBe) (Barnich and Van Droogenbroeck, 2011), and Pixel Based Adaptive Segmenter (PBAS) (Hofmann et al., 2012).

For example, a vehicle detection and classification system can be used by the road architects. The system is able to provide the number of vehicles and their classes using a particular stretch of the road. Road wear and safety are affected by the number of vehicles and their classes using the road. A large number of vehicle using the road will cause the road to wear faster. Heavy vehicles will also accelerate road wear. Hence, road architects need to take these information into consideration during planning and design stages. A more durable pavement type can be chosen for road with a lot of heavy vehicles. More frequent road maintenance can also be planned.

1.3 Problem Statements

The existing Vehicle Detection and Classification System developed by Lee Teng (Ng, 2014) does not have tracking and counting module implemented. Without tracking and counting module in place, the existing Vehicle Detection and Classification System is not suitable for practical use because of intensive manual work is required to count the vehicle. Tracking is required to enable automatic vehicle count because in a video sequences, a vehicle will

appear in multiple frames causing a single vehicle to be counted as many times as it appears in the video sequences frames. This makes counting impossible without a tracking mechanism. Random detection caused by noise can be easily filter out with tracking. This is due to the fact that this kind of noise usually appears in 1-2 frames only. Opposite traffic flows that is not desired can also be filtered out with tracking. By tracking a moving vehicle, the direction of the vehicle movement can be determined. Any vehicle that is moving upward will be ignored and not tracked.

1.4 Objectives of the Project

The objectives of this project include:

1. To develop and implement tracking and counting feature into the existing vehicle detection and classification system.

2. To assess the performance of the vehicle detection and classification with tracking and counting feature.

3. To select the optimal parameters for tracking and counting module.

1.5 The Scopes of the Project

The system which is developed in this project is able to detect, track, count and classify vehicle from traffic surveillance video sequences.

The input of the Vehicle Detection and Classification System developed in this project is limited to traffic surveillance video sequences that had been loosely calibrated. The angle of the traffic surveillance video camera sequences must be roughly pointing downward from an

overhead bridge. Video sequence that does not meet the requirement might still work but the system performance is going to suffer.

The traffic surveillance video sequences must be taken with good lighting condition at day time between 8am to 5pm. Video sequences taken during night time cannot be used. All video sequences are acquired in good weather condition without rain and fog. The output of the system is limited to total number of counted vehicles and also their classes (motorcycle, car, lorry, and non-vehicle).

1.6 Outline of Dissertation

Chapter one introduces the Vehicle Detection and Classification System. This chapter includes the problem statements, objectives of the project, significant of the project, scope of the project, and outline of the report.

Chapter two reviews the past researches and studies which contributed to the development of this Vehicle Detection and Classification System. Theories and knowledge in digital image processing is also discussed. Finally, the simple tracking and counting algorithm used in this system is explained.

Chapter three shows the hardware and software used for developing and running the system. The design overview is also discussed. Three of the main modules of the Vehicle Detection and Classification System are described.

Chapter four discusses the experimental results and performance of the system. The effects of tracking tolerance value and virtual line location value are evaluated. Final results of the vehicle classification is discussed and an overall system performance is given.

Chapter five concludes the project and state the future works. The limitation of the system is also discussed.

CHAPTER 2

LITERATURE REVIEW

2.1 Literature Review of Past Researches

Developing a Vehicle Detection and Classification System required three main components. There are detection algorithm, tracking and counting algorithm, and classification algorithm. The following sections will discuss these three components briefly.

2.1.1 Vehicle Detection

There are many methods to detect vehicles. These methods can be categorized into intrusive and non-intrusive as described by Daubaras and Zilys (2012). An example of intrusive method is installation of inductive loop detector into the pavement surface. Non-intrusive methods can be further categorized into imaging and non-imaging. Examples of non-imaging methods include using infrared sensor, microwave radar, and ultrasonic sensors. On the other hand, imaging methods include any use of digital image camera. Imaging method is a more preferable method because this method cost the least among non-intrusive methods as reported by Sun et al. (2004).

Two vehicle detection algorithms will be discussed. They are Visual Backgound Extractor (ViBE) and Pixel-based Adaptive Segmenter (PBAS).

Ng (2014) had done performance evaluation for Gaussian Mixture Model (GMM), PBAS, and ViBe. The outcome of the evaluation is PBAS outperforms ViBe in term of accuracy and

ViBe outperforms PBAS in term of speed. Hence, ViBe will be used as the vehicle detection algorithm in this project.

2.1.1(a) Visual Background Extractor (ViBe)

ViBe is a multi-model background subtraction algorithm with N background models that utilized conservative and randomized update policies. A pixel of background model will be randomly chosen to be updated once the corresponding pixel is detected as background. Barnich and Van Droogenbroeck (2011) reported promising results in term of Percentage of Correct Classification (PCC) for ViBe. Figure 2.1 shows the background model created using ViBe. Figure 2.2 shows the foreground model created using ViBe.

Figure 2.1: Background model created using ViBe.

2.1.1(b) Pixel-based Adaptive Segmenter (PBAS)

PBAS, similar to ViBe is also a multi-model background subtraction algorithm. Hofmann et al. (2012) reported that PBAS outperforms ViBe. However, PBAS requires more complex

Figure 2.2: Foreground model created using ViBe.

calculation compared to ViBe. This makes PBAS not attractive to real-time application.

2.1.2 Tracking And Counting

Tracking of moving objects from video sequences are usually done by tracking blobs (Magee, 2004). Blobs are collections of pixels that are connected together in an image. Hence, contours found from vehicle detection algorithm in this project can be used to track moving vehicles. This is because basically contours define blobs. The following shows a simple tracking and counting algorithm:

1. Capture current video sequences frame.

2. Find all detected moving objects contours.

3. Find nearest match of tracked object.

4. If nearest match found, then update the tracked object.

5. Else create a new tracked object.

6. Clean up tracked object list. Discard expired object.

7. Increment vehicle count counter when a vehicle is detected.

Figure 2.3 shows a tracked moving vehicles. The blue line shows the movement line of the vehicle.
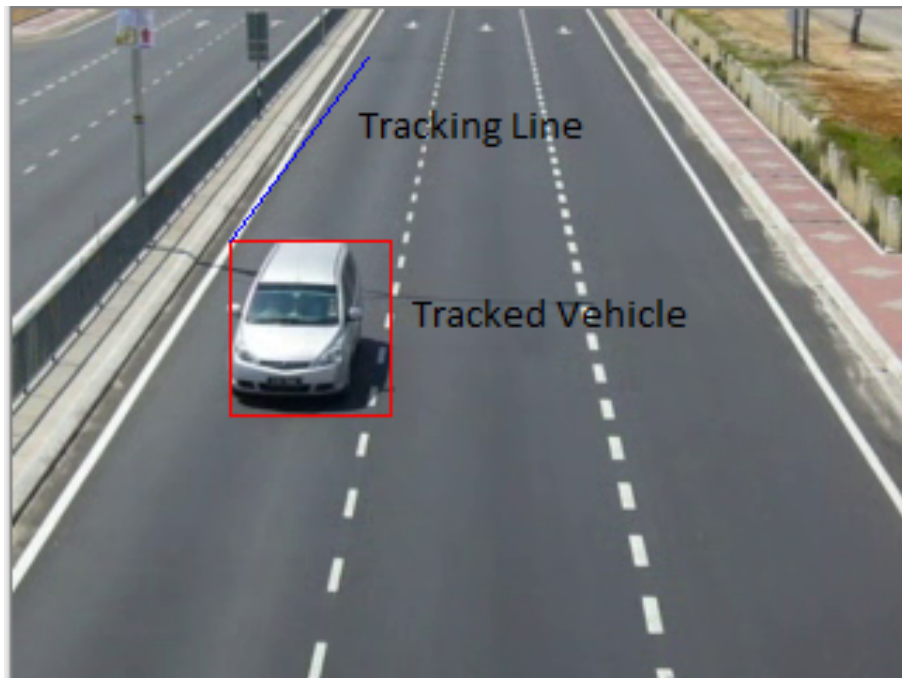

Figure 2.3: Tracked moving vehicle.

2.1.3 Vehicle Classification

There are a lot of classification methods such as K-Nearest Neighbour (KNN), Artificial Neural Network (ANN) (Freund and Schapire (1995) ; Ho and Tay (2008)), and Support Vector Machine (SVM). All these methods perform about the same depending on their applications. Therefore, SVM is used in this project since it is one of the most popular classification methods in traffic surveillance (Cheng et al. (2006); Sun et al. (2002); Rybski et al. (2010); Han et al. (2006)).

2.1.3(a) Histogram of Oriented Gradient (HOG)

Before a video sequence frame is sent to the classifier to be classified, its features need to be extracted. HOG is one of the popular feature extractor. Mao et al. (2010), Han et al. (2006), and Rybski et al. (2010) used HOG in their works. HOG has been reported by Teoh (2011) to outperform another popular feature extractor such as Gabor.

2.1.3(b) Multi-class Support Vector Machine (SVM)

SVM is a binary classifier that can only classify two classes. However, it can be modified to classify more than two classes. This is where the multi-class prefix comes in. SVM needs to be trained offline with positive and negative samples to create the classifier model database. HOG features are used in this project to train the classifier.

2.2 Digital Image Processing

Digital image processing is the use of computer algorithms to perform image processing on digital images. Digital image processing allows the use of much more complex algorithms for image processing, and hence can offer both more sophisticated performance at simple tasks, and the implementation of methods which would be impossible by analog means as stated in Rafael et al. (2002).

In particular, digital image processing is the only practical technology for:

1. Classification

2. Feature extraction

3. Pattern recognition

4. Projection

5. Multi-scale signal analysis

2.3 Image Enhancement

The purpose of the image enhancement is to process the image so that it becomes suitable for specific application. Image enhancement techniques include histogram equalization, spatial filtering, smoothing, sharpening, and contrast stretching.

2.3.1 Grayscale

A grayscale digital image is an image where each pixel from the image carries only the intensity information. Grayscale images are also known as black and white images. They actually contain different shades of gray ranging from the weakest intensity (black) to the strongest intensity (white). Figure 2.4 shows example of different shades of gray ranging from 0 to 255.
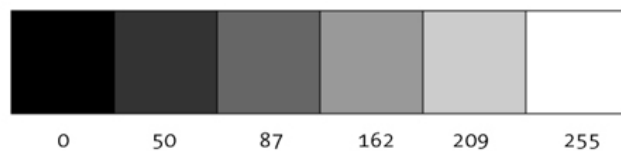

Figure 2.4: Example of Different Shades of Gray adopted from Processing.org.

There are many methods to convert color image to grayscale image. A popular method is to match the luminance of the grayscale image to the luminance of the color image. Different weighting of the color from the color image are normally used. The red, green, and blue (RGB) are normally weighted at 30%, 59%, and 11%, respectively. These weights depend on application but the given weight in the example are typical weight used. Another popular weight is given by Equation 2.1 where R, G, and B are red, green, and blue, respectively.

$$GrayscaleIntensity = \frac{(11*R+16*G+5*B)}{32} \qquad (2.1)$$

It is also possible to convert grayscale image back to RGB image by setting all the three primary color components red, green, and blue to the gray value and adjust the gamma accordingly.

2.3.2 Morphological Image Processing

Morphological image processing describes a range of image processing techniques that deal with shape of features in an image. Among the morphological image processing include dilation, erosion, opening, closing, Top Hat, and Black Hat.

Table 2.1: Basic Morphological Operator adopted from Choo (2009).

Operator	Mathematical Equation	Descriptions
Dilation	$A \oplus B$	A is dilated by structuring element B. Boundary of A expanded.
Erosion	$A \ominus B$	A is eroded by structuring element B. Boundary of A contracted.
Opening	$A \circ B = (A \ominus B) \oplus B$	A is eroded first and then dilated by structuring element B. Eliminates false touching of two objects. Separate two adjacent objects.
Closing	$A \bullet B = (A \oplus B) \ominus B$	A is dilated first and then eroded by structuring element B. Fills small gaps and holes. Connects two adjacent objects together.
Top-hat	$A \ominus (A \circ B)$	Extracts feature which is smaller than structuring element B and is brighter than surroundings.
Black-hat	$(A \bullet B) \ominus A$	Extracts feature which is smaller than structuring element B and is darker than surroundings.

2.4 Summary of Vehicle Detection and Classification Method

Generally, popular vehicle detection methods are Gaussian Mixture Model (GMM), Visual Backgound Extractor (ViBE) and Pixel-based Adaptive Segmenter (PBAS). Each of the methods perform well in different scenario. For example, ViBe outperforms PBAS in term of processing speed because PBAS requires more complex calculation compared to ViBe. Ng (2014) had done a comprehensive comparison of these three methods with ViBe showing promising results. Hence, ViBe is also chosen to be used in this project.

SVM is chosen to be used in this project as the classification method. This is because SVM is a proven popular classification method. Ng (2014), Cheng et al. (2006), Sun et al. (2002), Rybski et al. (2010), and Han et al. (2006) have all used SVM in their works.

2.5 Summary of Tracking and Counting Method

Blob tracking method is chosen to be used in this project. Extensive work on blob tracking for vehicle application are done by Magee (2004). A modified version of blob tracking is used in this project due to ease of implementation.

CHAPTER 3

METHODOLOGY

3.1 Hardware and Software Used

The vehicle detection and classification system consists of hardware and software components that process an input graphical signals such as video sequences from traffic surveillance camera into the number of vehicle and their type. The hardware required for this system includes a computer and a camera that are able to capture video sequences with at less 320x240 resolution at 25-30 frames per second (FPS). The software part of the system is developed using Microsoft Visual Studio with OpenCV library by IntelCorporation (2001).

3.1.1 Hardware

In this project, the system is developed and run on a laptop. The laptop has Intel Core i5-4300U (1.90 GHz) processor with 4 GB of RAM running Windows 8.1 Operating System (OS). The traffic surveillance video sequences were taken using normal digital camera and also surveillance camera.

3.1.2 Software

The Vehicle Detection and Classification System software is developed using Microsoft Visual Studio 2010 IDE (Integrated Development Environment) with OpenCV (Open Source Computer Vision) library to simplify the programming task.

3.2 Design Overview

The design of the Vehicle Detection and Classification System can be divided into three modules. The first module is for detection of the vehicle. The second module is for tracking and counting. The last module is for classification of the vehicle type. Since the detection and classification modules already exist, they are going to be reused. Hence, the focus of this project is to develop and implement the tracking and counting module into the existing system. Figure 3.1 shows the block diagram of the Vehicle Detection and Classification System.
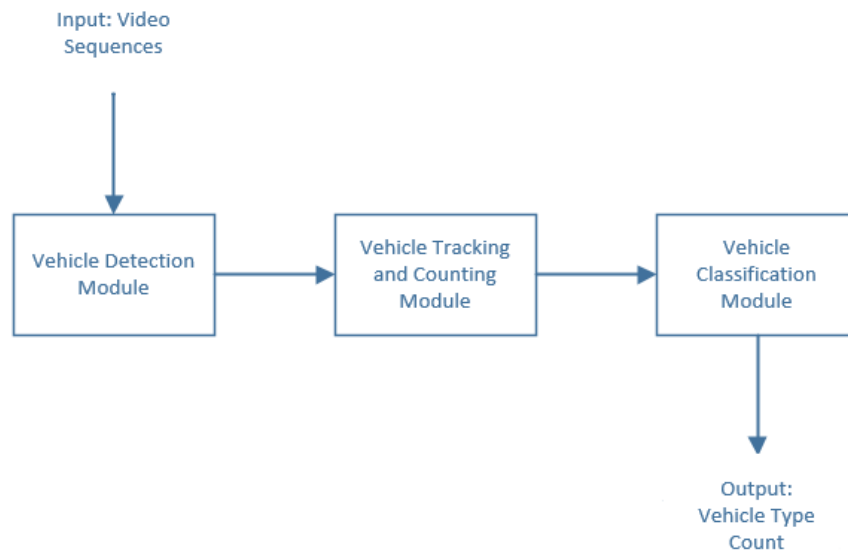
Figure 3.1: Block Diagram of the Vehicle Detection and Classification System.

The input of the system is video sequences of traffic. It can be either taken using normal digital camera or traffic surveillance camera. However, the video sequence must be loosely calibrated where the angle must be pointing downward and the road width must be in sight.

The video sequences will then be processed frame by frame. Pre-image processing will be applied before the frame is sent to vehicle detection module. Background is removed by the

vehicle detection module and the foreground is sent to the tracking module in contour format.

Vehicle tracking and counting module will label all the contour information received from the vehicle detection module. It will find the nearest match and updated the tracking information accordingly. If no match is found, the particular object will be tracked as new object.

When the tracked vehicle passed through a pre-determine virtual line, the tracked vehicle will be sent to classification module to classify its' type. The counter of the identified vehicle type is incremented accordingly.

Finally, the system will output the count of each vehicle type and also the total number of vehicle counted.

3.3 Vehicle Detection Module

Vehicle detection module function is to detect vehicle from the given video sequences and extract the vehicle to be processed by vehicle tracking and counting module and vehicle classification module. Figure 3.2 shows the flowchart of the vehicle detection module.

Pre-image processing is applied to video sequences frames before being sent to ViBe. ViBe will extract the background and foreground of the given frame. After that, some morphological operation is done before removing the shadow. Shadow removal is then followed by contour extraction. Lastly, the extracted contours information will be sent to vehicle tracking and counting module.
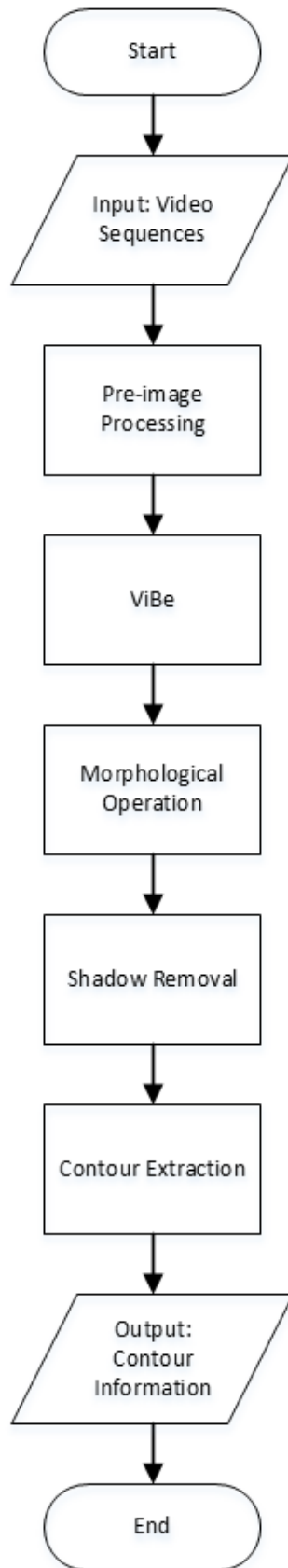
Figure 3.2: Flowchart of the vehicle detection module.

3.4 Vehicle Tracking and Counting Module

Vehicle tracking and counting module tracks detected object from the video sequences. Figure 3.3 shows the flowchart in the vehicle tracking and counting module.

Vehicle tracking and counting module takes the contour information from vehicle detection module to find the nearest match in the tracked objects list. The purpose of finding nearest match is to decide whether the given contour had been tracked previously. Nearest match is found by comparing the contour position and contour size from each of the tracked objects. Criteria of match is based on the contour position and the contour size that are within the tracking tolerance value. If there is a match, the tracked object will be updated accordingly. If there is no match, a new tracked object will be created. After all the contours information from a video sequences frame is processed, the vehicle tracking and counting module will clean up its tracking list. Tracked object with expired timer will be discarded. Tracked object that has already crossed the pre-determine virtual line will be counted as vehicle detected. Virtual line location with 120 pixels at y axis will be used initially. This value is chosen because it divides the frame equally into half. An optimal value of virtual line location will be determined base on experimental result. This tracked vehicle information will be sent to vehicle classification module for vehicle type classification.

Important parameters for tracking and counting module:

1. Tracking Tolerance Value - a tolerance value used by the tracking and counting algorithm to find the nearest match of the tracked object. If the tracking tolerance value is too small, there will be a lot of false mismatch. On the other hand, if the tracking tolerance value is too huge, there will be a lot of false match.

2. Virtual Line Location - a virtual line drawn across the y axis of the video sequences
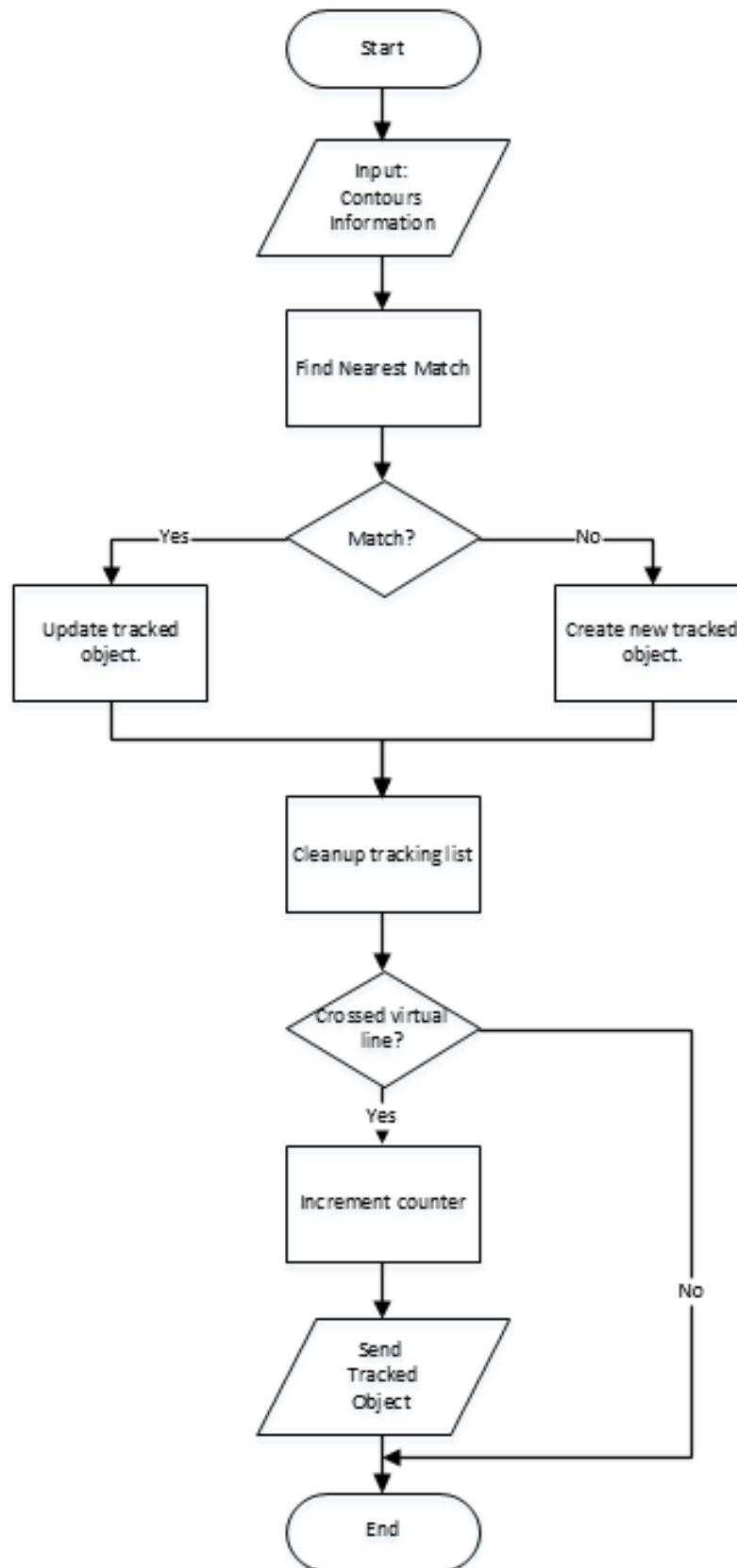
Figure 3.3: Flowchart of the vehicle tracking and counting module.

where tracked objects that pass through this line will be counted as detected vehicle and then classified to their respective type of classes.

3. Tracking Timeout Value - determines when to discard a tracked object. Value of 1 means the tracked object will be discarded if it has not been updated in 2 subsequence frames.

4. Occurrence Threshold Value - determines when a tracked object is marked as potential candidates to be counted as vehicle. This will affect how much noise will be filtered out. The smaller the threshold value, the less tracked object will be filtered out and via versa.

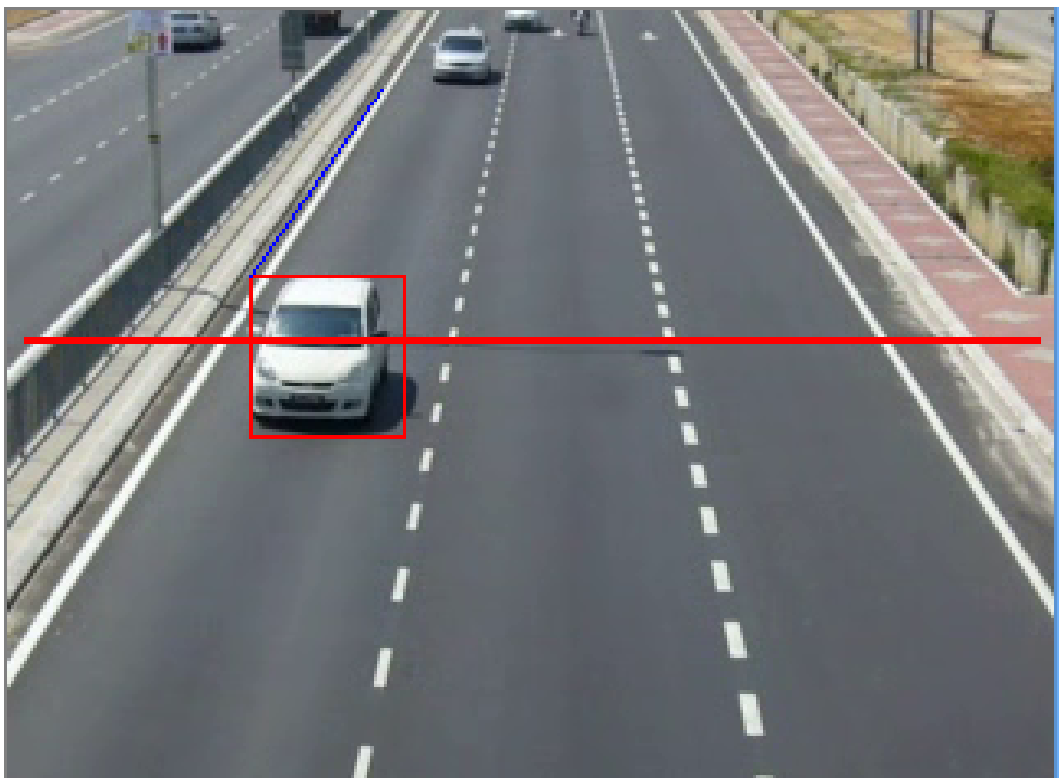Figure 3.4 shows an examples of virtual line across the *y* axis.

Figure 3.4: Virtual Line Across (red line) the *y* axis.

3.5 Vehicle Classification Module

This is the last module of the Vehicle Detection and Classification System. The function of this module is to classify the vehicle into their respective classes. Figure 3.5 shows the flowchart of the vehicle classification module.
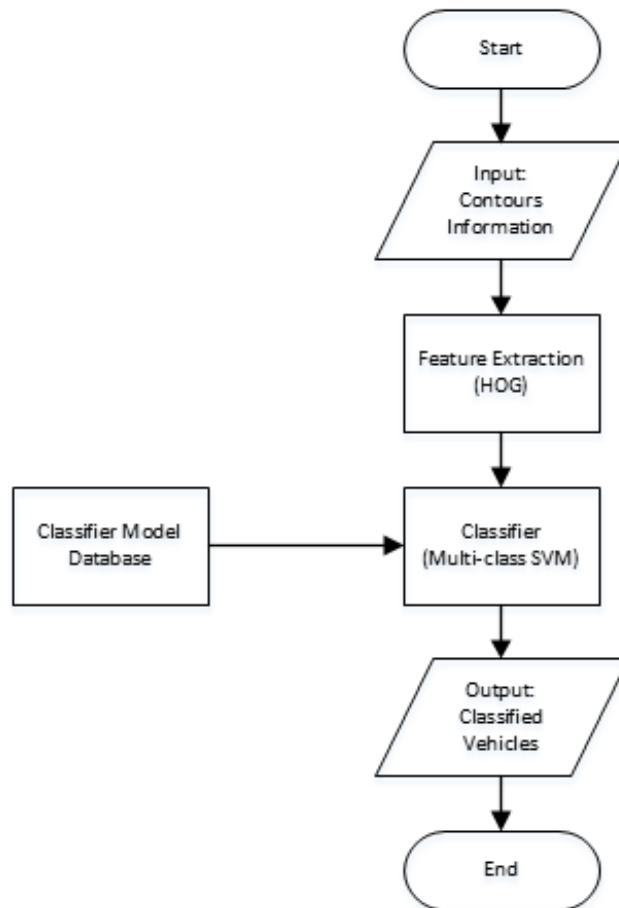

Figure 3.5: Flowchart of the vehicle classification module.

Extracted foreground processed frame is sent to the vehicle classification for HOG feature extraction. The extracted features are then sent to the classifier to have the vehicle classified. The classifier uses offline trained database model to classify the vehicles.

Positive images of each class of the vehicle are used to train the classifier. In this project,

the positive images will be motorcycle, car, lorry, and non vehicle images. Feature extraction using HOG is used in this project. The extracted features are then used to train the classifier and a classifier model's database is built for classification use later on.

3.6 Performance Evaluation Method

The performance of the tracking and counting module and the overall system performance are evaluated by varying tracking tolerance value and virtual line location.

3.6.1 Tracking Tolerance Value Evaluation Method

Tracking tolerance value affects vehicle detection performance. When finding the nearest match, the tolerance value determines whether the given contour belongs to tracked vehicle or untracked vehicle. In order to determine the optimized tracking tolerance value, a range of tracking tolerance value from 5-30 is evaluated. For this evaluation, the other parameters are arbitrary fixed. For example, tracking timeout value is set to 1 frame, occurrence threshold value is set to 3 frames, and virtual line location is set at 120 pixels at y axis. Descriptions of these parameters can be found in Section 3.4.

Percentage of Detection (PD)is calculated by Equation 3.1 while Percentage of Correct Detection (PCD)is calculated by Equation 3.2.

$$PD = \frac{NumberOfCorrectDetection}{TotalVehicle} * 100\% \tag{3.1}$$

$$PCD = \frac{NumberOfCorrectDetection}{NumberOfVehicleDetected} * 100\% \tag{3.2}$$

PD is used instead of PCD because PD is able to show the performance of the tracking and counting module clearly. For example, when tracking tolerance value is varied, the PD varied accordingly with obvious differences while PCD only varied slightly.

3.6.2 Virtual Line Location Evaluation Method

Virtual line location also affects vehicle detection performance. Virtual line location used in section 4.2 was 120 pixels at *y* axis. This value is chosen arbitrarily previously as it divides the video sequences frames equally into half. Virtual line location will be varied from 80-130 pixels along *y* axis to evaluate the tracking and counting module performance.

3.6.3 Overall System Performance Evaluation Method

The overall system performance is evaluated by calculating the Percentage of Correct Classification (PCC), False Positive Rate (FPR), and False Negative Rate (FNR).

Percentage of Correct Classification (PCC) is calculated by Equation 3.3.

$$PCC = \frac{NumberOfCorrectClassifiedVehicles}{TotalVehicleInSpecifiedClass} * 100\% \qquad (3.3)$$

False Positive Rate (FPR) is calculated by Equation 3.4.

$$FPR = \frac{FalsePositive}{FalsePositive + TrueNegative} * 100\% \qquad (3.4)$$

False Negative Rate (FNR) is calculated by Equation 3.5.

$$FNR = \frac{FalseNegative}{TruePositive + FalseNegative} * 100\% \qquad (3.5)$$