PEDESTRIAN COUNTING SYSTEM

ANDY SIM

A project report submitted in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic Engineering

Faculty of Engineering and Science
Universiti Tunku Abdul Rahman

September 2016

DECLARATION

I hereby declare that this project report is based on my original work except for citations and quotations which have been duly acknowledged. I also declare that it has not been previously and concurrently submitted for any other degree or award at UTAR or other institutions.

Signature : _____

Name : ANDY SIM

ID No. : 11UEB02823

Date : _____

APPROVAL FOR SUBMISSION

I certify that this project report entitled **"PEDESTRIAN COUNTING SYSTEM"** was prepared by **ANDY SIM** has met the required standard for submission in partial fulfilment of the requirements for the award of Bachelor of Engineering (Hons.) Electrical and Electronic Engineering at University Tunku Abdul Rahman.

Approved by,

Signature : _____

Supervisor : _____

Date : _____

Specially dedicated to
my beloved mother, father and Mr Ng Choon Boon.

ACKNOWLEDGEMENTS

I would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Mr. Ng Choon Boon for his invaluable advice, guidance and his enormous patience throughout the development of the research.

In addition, I would also like to express my gratitude to my loving parent and friends who had helped and given me encouragement and information in proceeding my research.

PEDESTRIAN COUNTING SYSTEM

ABSTRACT

Real-time pedestrian counting is on high demand for traffic control, marketing and management of the crowd. In this project, an automatic and robust pedestrian counting system for a shopping mall based on computer vision is proposed. It is very important to determine the recession and peak hours of the mall by visitors counting for business decision making. The proposed method utilizes a pre-installed overhead or surveillance camera on the entrance of the sidewalk or a corridor in the mall compound for security purpose. By accessing the camera, the area of interest is scanned, images are processed and the pedestrians in the regions are detected. By using different approach of image and video processing methods, this system can be implemented in Visual Studio with the help of OpenCV image processing libraries. Several methods of image processing techniques such as background subtraction, optical flow, Harr-like feature detector and Histogram of Gradient are reviewed in this report. A pedestrian counting method based on Histogram of Gradient is presented. The principle of HOG is introduced and the detection accuracy, parameters of configuring the system and also the defectiveness of the system is being discussed. The proposed system is able to detect the positions of human accordingly to real time. The computer experiment has shown that the system is able to detect and count the human correctly.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

β	weight update parameter
G_t	background image at time t
$\mu_{m,t}$	mean
$\sigma^2_{m,t}$	variance
a	weight update parameter
$w_{m,t}$	weighting coefficient
v	specific volume, m^3
I_{rx}	I_r in x direction
I_{gr}	grey scale image
I_{grx}	grey scale image at x direction
$I(x,y,t)$	grey value of the image I at pixel (x,y)
I_o	original image
I_r	R channel for I_o
T^V	difference between their decrease
T^V	difference between their decrease
$B_k^H(x,y)$	hue channel of pixel (x,y) in the background frame
$C_k^H(x,y)$	hue channel of pixel (x,y) in the current frame
$B_k^S(x,y)$	saturation channel of pixel (x,y) in the background frame

$C_k^S(x,y)$	saturation channel of pixel (x,y) in the current frame
$B_k^V(x,y)$	value channel of pixel (x,y) in the background frame
$B_k^V(x,y)$	value channel of pixel (x,y) in the background frame
$C_k^V(x,y)$	value channel of pixel (x,y) in the current frame
T	pre-set threshold

C0	first criterion
C1	second criterion
CCTV	closed-circuit television
DVR	digital video recorder
HOG	histogram of gradient
HSV	hue, saturation and value
ID	identity
IP	internet protocol
PCA	principal component analysis
RGB	red-green-clue
RJ45	modular connector used for Ethernet computer network cable
ROI	region of interest
SVM	support vector machine
US	United State
USB	Universal Serial Bus

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Background

As far as the human history goes, crimes never cease to exist. Therefore, surveillance is introduced. Surveillance, the word originated from French early 19th century means keep watch in Latin has been used quite frequent after the year 1950. In 1980's, the very first generation of commercial video surveillance system which is the analogue closed-circuit television (CCTV) was developed. The video camera is connected to surveillance monitors by coaxial cables for monitoring purpose by human operators. In 1990's, the video surveillance system is improved and replaced by videotape digital video recorder (DVR).

Moving on, the third generation of the video surveillance system was introduced as an IP network system. The new system stored data continuously in servers or hard disk drives after being transmitted over the network. Back in 1940's, Germany had already development the CCTV system for observing the launch of V2 rockets. Besides that, US military had also developed CCTV for atomic weapon testing at safe distance.

Surveillance or overhead cameras are used for the purpose of observing, managing and protection. As the blooming of domestic economic development goes, many large stores, hypermarkets, supermarkets and public transport stations are always crowded with people. Therefore, human resources department must provide a better plan to ease the traffic conjunction.

Number of customer is indispensable data for making business decisions. The information of customer flows is used as an index for shopping stores deployment and resources allocation. Normally, the number of customers per year is used to predict the peak periods of the company and stores. In the exhibition hall or stadium, we can determine the number of visitor entering the premises and direct the information for the security department. From there, they would know how much security force is needed and at the same time, controlling the traffic flow.

Although the traditional manual counting such as revolving door clickers has very high accuracy, higher labour cost and resources are needed (Ryan et al., 2007). In addition, they are not portable and use up a lot of space. In peak hour, they will become a barricade to block the traffic and slow down the entire flow of people entering. This is not wise as they tend to affect the emotional level of the customers due to long queuing time. Therefore, in the current society, surveillance or overhead camera video counting system is introduced. People counting system is an automatic system that count the number of people in the enclosed or open areas (Xi et al., 2009). With computer vision techniques, the intelligent surveillance camera is able to perform more meaningful actions such as detecting events and people tracking The system is able to track and capture the motion information of the targets for high-level analysis.

1.2 Aims and Objectives

In this project, a pedestrian counting system for shopping mall was proposed. According to the existing solutions, a pedestrian counting problem is developed. Some video analytic skills are being performed to analyse video feed such as in the application of object detection, monitoring and tracking. The work load of human operator is being reduced and the time consuming for error correction could also being reduced with this technology as it could assist the human in making decision.

In this project, video detection system is being used for pedestrian counting. With the used of an overhead or surveillance camera, people counting is possible

with the help of computer vision technologies. A video is fetch from the camera through IP or Ethernet cable and also USB to the computer and then process by a robust and adaptive algorithm which is developed for excellent counting accuracy. Then the frame is extracted from the video and process with the help of image processing algorithm. The video-based pedestrian counting system is being developed on the Windows 10 platform with Microsoft Visual Studio 2015 along with the OpenCV development tools.

1.3 Gantt Chart

Table 1.1: Schedule of FYP 1

Week	Task	Week Duration	Start Week
Week 1	FYP Title Registration and Consult Supervisor	1	1
Week 3	Journal Reading	5	2
Week 8	Journal Reading & Chapter 1	1	7
Week 9	Journal Reading & Chapter 2	4	8
Week 12	Journal Reading & Chapter 3	1	12
Week 13	Finalizing the FYP Report	1	13

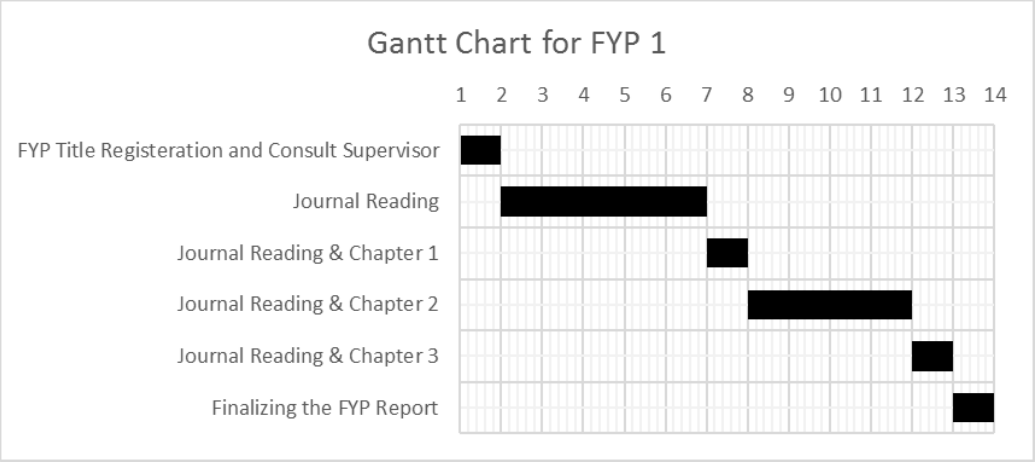

Figure 1.1: Gantt Chart for FYP 1

Table 1.2: Schedule of FYP 2

Week	Task	Week Duration	Start Week
Week 1-12	Coding for Pedestrian Counting System	12	1
Week 13	Finalizing the FYP Report	1	13

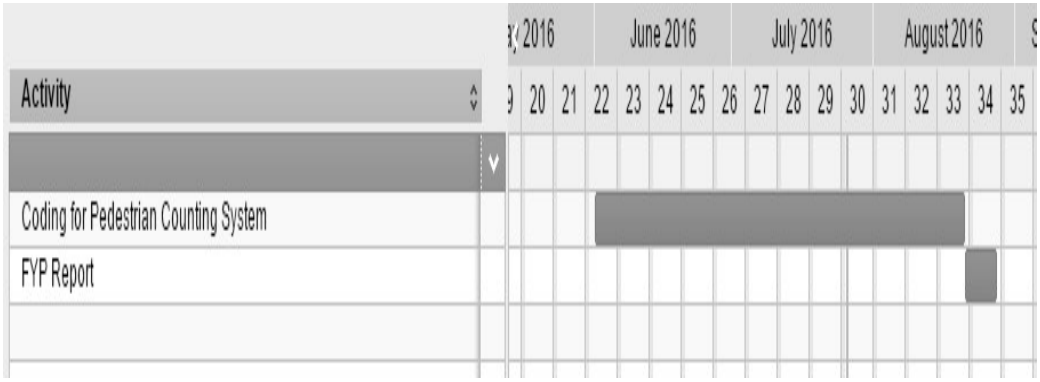

Figure 1.2: Gantt Chart for FYP 2

CHAPTER 2

LITERATURE REVIEW

2.1 Pedestrian Counting System

Pedestrian counting system is being classified into two categories in Kamel et al. (2004), obstructive and non-obstructive. Obstructive people counting system are turnstiles and mat-type foot switched where they have the tendency to slow down the traffic as it can obstruct the passage way and at the same time, high cost and low flexibility as it was installed on the ground (H.Hakan et al., 2014). Whereas for the sensor type counting system, is considered as non-obstructive systems where they do not affect the traffic but has the tendency of undercounting (Senem et al., 2006). On the other hand, computer vision technologies which is also one of the non-obstructive system, has higher accuracy as it is able to identified multiple target at the same time and less bothersome in setting up the system (Keitaro, 2012).

2.2 Working Principle of the Pedestrian Counting System

A simple and generalized pedestrian counting system is as follow:

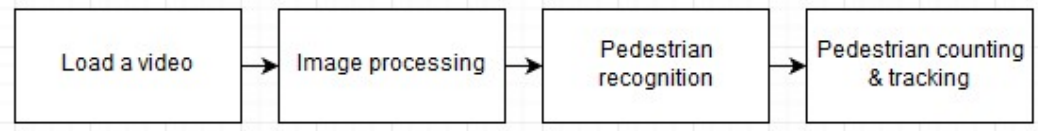

Figure 2.1: System block diagram

In the paper of Damien (2007), in order for the pedestrian counting system to work, people must be separated from a background scene. By determining the foreground and background object, people are able to be determined as a target by the system. Foreground and background processing which are based on the colour texture, shadow and lightning condition are performed with the computer vision technologies based methods. With the classifier, the system is able to recognize and differentiate between human and non-human object through machine learning and face detection methods. With other image processing method, the image is processed so that it is always easier to be classified and at the same time, reduce the computer spaces and computation time.

Many methods have been proposed for pedestrian detection. Like Terada et al. (1999), they had created a method where people were being detected as they crossed a virtual line. This method is able to avoid occlusion which was a real headache problem in image processing. But, in order to determine people's direction, a space-time image is required. Onwards, Hashitmoto et al. (1997) resolve the problem of people counting with background subtraction to create thermal images and region of interest. The system proposed had an accuracy rate of 95%.

Later on, Haritaoglu et al. (2001) developed another method which is the real-time tracking. They improved the background subtraction method with colour and intensity of pixel values. This information would contribute to the classifying of pixels of the images, such as foreground, background and shadow. Finally, Gary Conrad and Richard Johnsonbaugh had simplified the whole system by using an overhead camera with consecutive frames differencing. This method was adopted instead of background subtraction to avoid light modification, occlusion. With the theory behind, they successfully developed a simple algorithm that had a 95.16% accuracy rate. Several methods are being discussed below.

2.2.1 Background Subtraction and Segmentation Method

According to Chinchkhede et al. (2012), background subtraction method must be able to adapt the fast changing or lightning and illumination. In Damien (2007) paper, it is stated that background subtraction will not remove all the wrong pixels and occlusion has a great impact on it. However, this method was accepted by major people as it is very difficult to separate the foreground and background of the image by using the frame differencing process as it has the inclination to highlight only the edges of the objects in the foreground. Hence, the image analysis will be much more complex and difficult. The con of background subtraction method located in its ineffectiveness of detecting stationary objects of the foreground. In this perspective, frame differencing was much better at it. With the proposed method, noises such as shadow would be eliminated. However, remnants of noises could still be seen from the processed image. Mathematical morphology such as erosion and dilation were introduced to remove the remnants noises.

Erosion, morphological operator that was used to remove the boundaries of the different foreground regions, resulting the objects becoming smaller and holes to be larger.

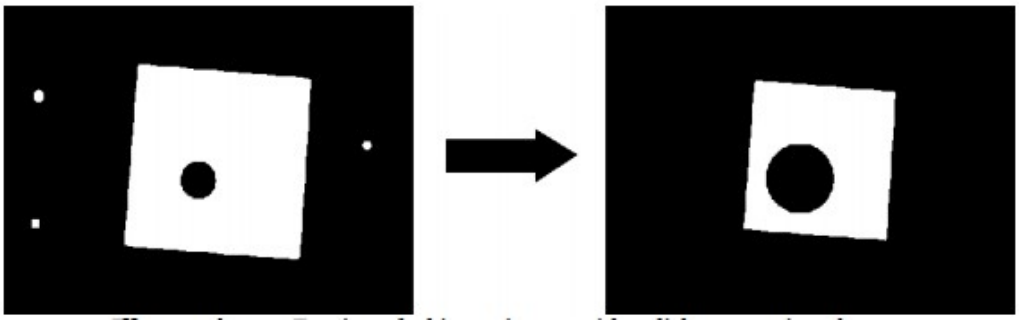

Figure 2.2: Erosion of binary image (Damien, 2007)

In order to compute a binary erosion, pixels of the foreground must be processed. For each pixel, the algorithm will put the structure element and test them if there were completely contained in the foreground or not. If it wasn't, the current

pixel will be judged as background. On the other hand, if the result is positive, the current pixel will be enclosed in the eroded foreground.

Dilation, the second morphological operation that dilate the boundaries of the foreground regions. As a result, the forefround object becomed larger while the holes become smaller as shown in figure below.

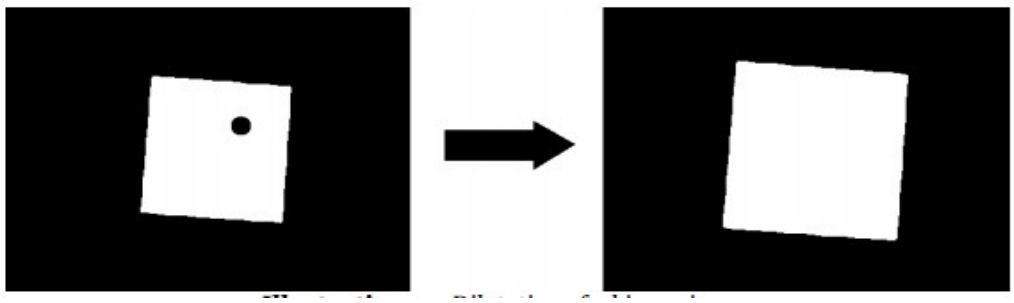

Figure 2.3: Dilation of binary image (Damien, 2007)

Lastly will be the opening morphological operation which made use of erosion and dilation. From the figure below, we can see that all pixels of an images which were contained by the structuring element would be preserved within the forground entirely. However, all foreground pixels which cannot be enclosed by the sturcturing element would eroded away.

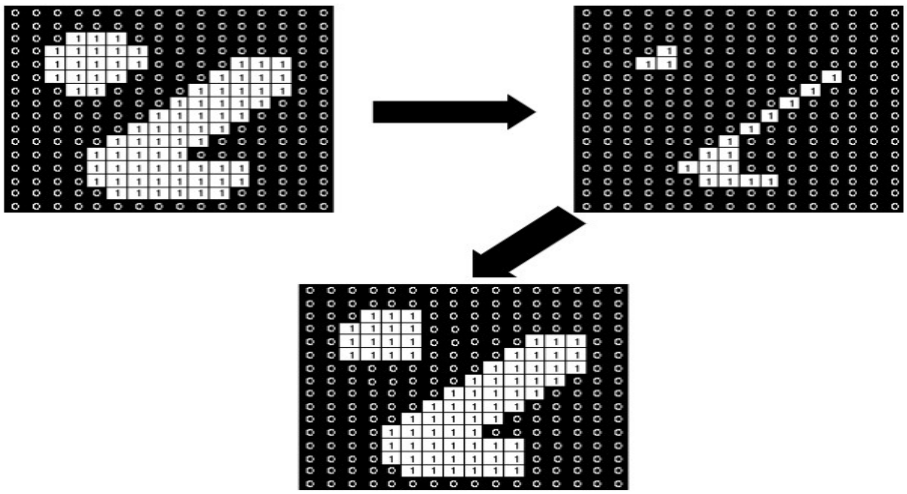

Figure 2.4: Opening of binary image (Damien 2007)

2.2.2 Frame Differencing Method

As mentioned by Damien (2007), background subtraction often had to deal with the occlusion, frame differencing was developed to solve that problem. By making pixel by pixel absolute difference between two consecutive frames, the first step of people counting was completed. The result of the differencing was that a new image was created that showed all the differences between two consecutive frame that acts as a motion detector.

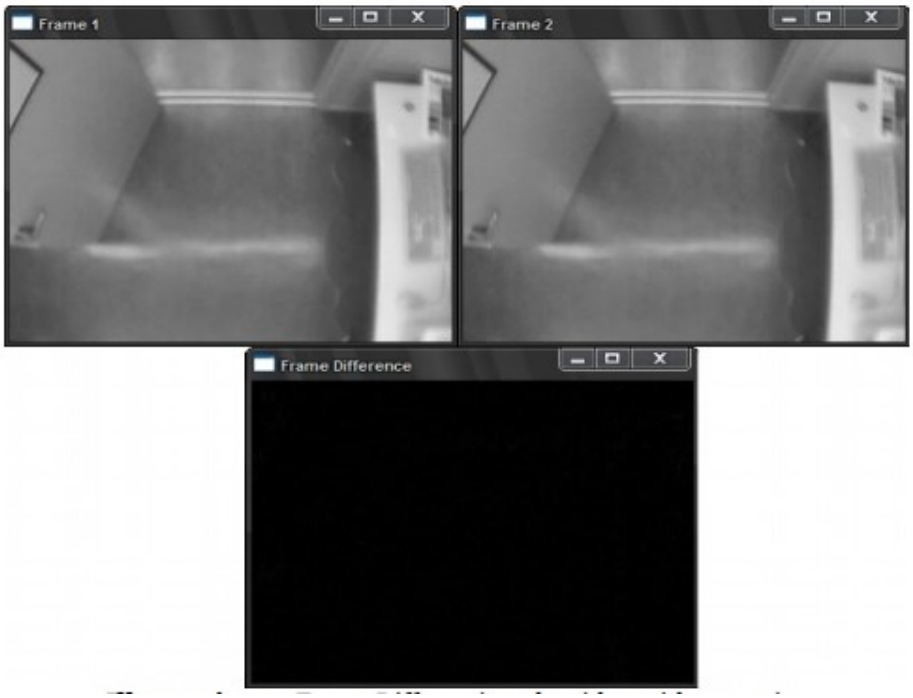

Figure 2.5: Frame differencing method without motion (Damien 2007)

Figure 2.6: Frame differencing method with motion (Damien 2007)

Due to the quality of camera, noises was generated. The noises and some automatic processing system by the camera, the result of frames differencing will neither be zero, nor too big. Therefore, a threshold was implemented to decide the existense of motion. There were two ways that the threshold could be done. Firstly, a threshold between each pixel. All difference value smaller than a predecided value after trial and error will be judged as a null pixel (black). Then the maximum gray-scale value was taken and compared to the threshold. If the value is bigger than the threshold value, the there is motion.

Secondly, threshold in full difference image where the sum of all pixel values must be greater than the threshold value, otherwise it would be considered as empty images. The only problem here was the threshold value. In different lightning condition, the threshold value varied. Therefore, the threshold value could only be determined via try and error.

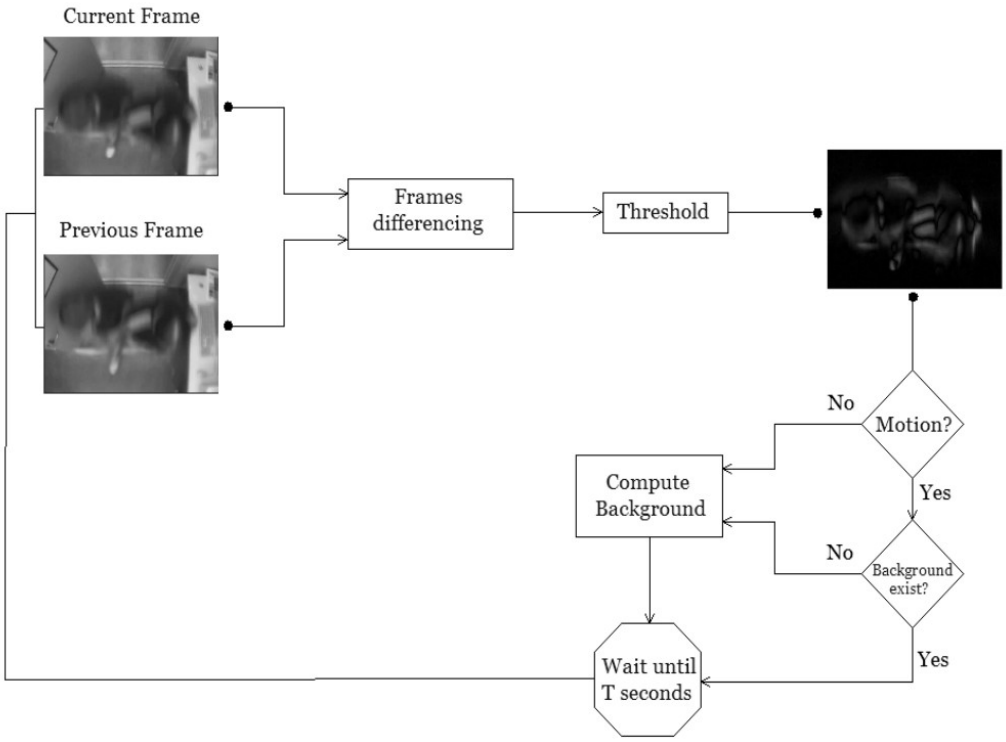

Figure 2.7: Background estimation module (Damien 2007)

2.2.3 Extracting Foreground Objects Method

Image processing normally starts off with an edge detection process then only followed by feature extraction process. This is to reduce the amount of information in the image. Xun et al, (2013) also supported the conclusion of Damien (2007) in which the frame differencing is a better and more adaptive method compared to the background subtraction. However, background subtraction and frame differencing are not good enough to detect foreground due to their sensitivity of lightning condition. Therefore, a representation of the background for the frames is a more advanced approach.

Although there are better models other than Gaussian mixture model such as the parametric methods proposed by Elgammal et al. (2000) and Oliver et al. (2000), Gaussian mixture model is more commonly been used due to its ease of use and the implementation of Kalman filter which is very compatible to the model. YuanYuan el al. (2013) had introduced a background modelling method which was the Gaussian

mixture modelling. By representing the characteristic of the pixel in the frame or images with the Gaussian probability density function, the Gaussian mixture model for each pixel at time t,

$$P(G_t) = \Sigma_{m=1}^{k}(\omega_{m,t} f(G_t, \mu_{m,t}, \sigma_{m,t}^2) \quad (2.1)$$

was created for determining whether the object was a foreground point or a background point. As the time changed, the background image also changed as the Gaussian model was constantly updated.

$$w_{m,t+1} = (1+\alpha) w_{m,t}, \mu_{m,t+1} = \beta \mu_{m,t} + (1-\beta) I(x,y,t) \quad (2.2)$$

Then, a pre-set threshold,

$$e^{\frac{-(I(x,y)-\mu(x,y))^2}{2\sigma^2(x,y)}} > T \quad (2.3)$$

was being implemented as a judge for determining the objects. If the equation was satisfied, (x, y) was determined as a background point, if not, then considered as a foreground point.

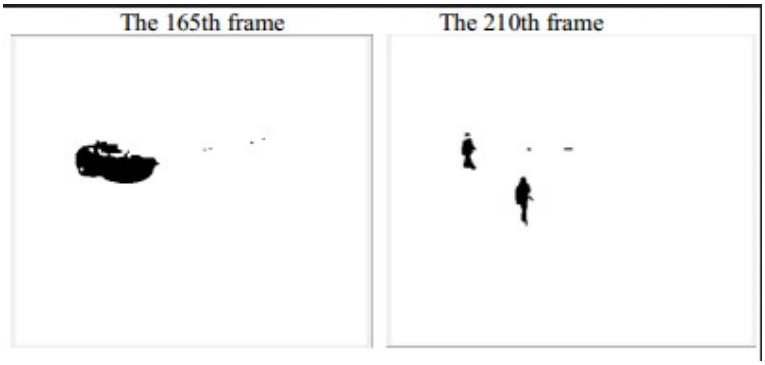

Figure 2.8: Foreground extracted by background subtraction method (Chen et al., 2013)

2.2.4 Screening Foreground Objects Method

Traditionally, foreground object was separated from the image by detecting the most distinct shape features such as aspect ratio. This method can efficiently remove fluctuation large objects such as trolley. These type of method had low accuracy. Therefore, a classifier was introduced to improve the accuracy of detection. In Yuanyuan et al. (2013) paper, the video frame is divided into 3 regions compile of horizontal or vertical axis $(0, x_0), (x_0, x_1), (x_1, x_2)$ with condition

$$\begin{cases} 0.001 < \frac{\delta}{s} < 0.01, if\ 0 \leq x_0 \\ 0.002 < \frac{\delta}{s} < 0.05, if\ x_0 \leq x_1 \\ 0.004 < \frac{\delta}{s} < 0.1, if\ x_1 \leq x_2 \end{cases} \tag{2.4}$$

If above statement is satisfied, object that is too large or too small will be removed. Furthermore, pedestrian was able to be determined if the aspect ratio of the object bounding rectangle satisfied the equation,

$$0.2 < scale = \frac{width}{height} < 0.8 \tag{2.5}$$

In addition, classification was implement for pedestrian detection. Each object was treated as a separated region. Then, normalization and HOG feature extraction were performed on the object with the support vector machine (SVM) and the Gaussian kernel as the classifier. SVM is a well-known classifier used in pedestrian counting decision making process. The SVM is trained to classify a pattern through learning by accepting a variety of samples and formulates classify the similarity or determining the relationship between the samples.

2.2.5 Haar-Like Based Pedestrian Detection and AdaBoost Classifier Method

In the paper of Chen et al. (2014), it was stated that all the previous work for pedestrian detection could be classified into two groups by camera view. The first one captured the video clips by long-shot with non-direct downward camera while the other by short-short with direct downward camera. The image processing method

used in the first group is to detect and track the foreground group by body features and segment them into single object for counting. This method had higher computational complexity if compared to the second group as the first group doesn't deal with occlusions. On the other hand, the second group method was to detect foreground object and segment them into head-shoulder shape featured individuals. This method had a smaller viewing angle if compared to the first group method.

However, they were not suitable for the pedestrian counting system in the supermarket due to the non-direct downward view. It was always difficult to segment groups into a single entity due to merging and body shape deformation. Therefore, Haar-like feature was introduced. Harr-like based detection involved training and recognition process.

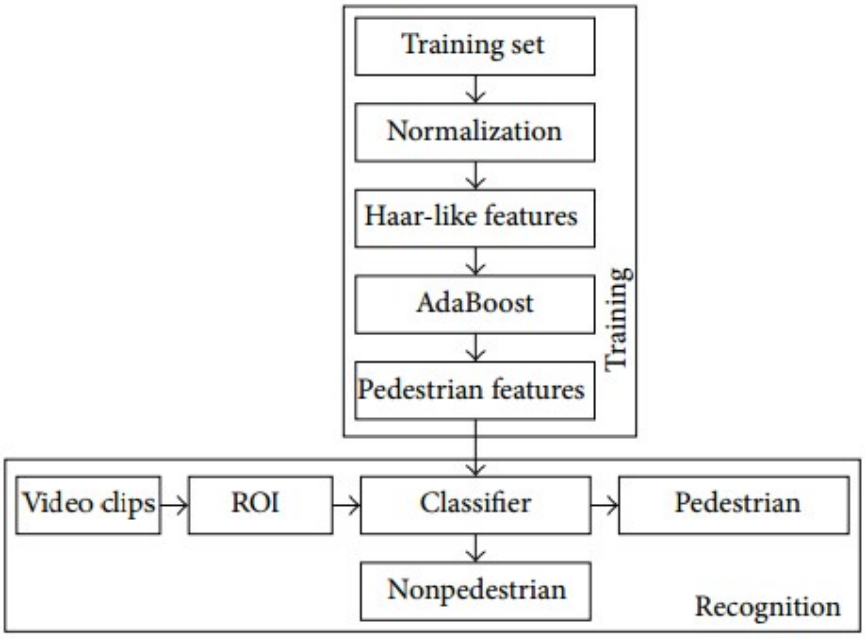

Figure 2.9: Flowchart of Haar-like based pedestrian detection (Chen et al., 2014)

As mentioned before, the algorithm was trained to recognize and differentiate object from target.

Figure 2.10: Pedestrian training software (Chen et al., 2014)

Therefore, positive samples and negative samples such as images of human head, mountain, river, cartoon, animal and etc were fed into the system or more correctly the classifier. By doing so, the classifier was trained to judge whether the object is a head of human or non-human. Initially, image samples' raw pixel value were used to train a classifier, but it was rejected due to the calculation complexity if actually put into use. Therefore, Harr-like feature consisted of 2 or 3 white and black colour rectangles joining together was chosen instead of the raw pixel value as shown in figure below.

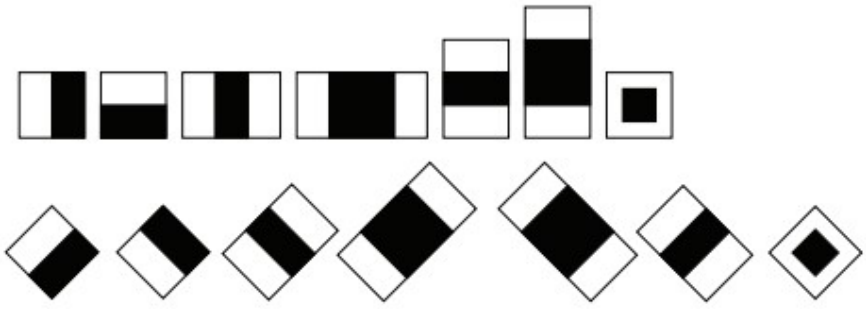

Figure 2.11: Haar-like feature prototypes (Chen et al., 2014)

After obtaining the Haar-like features value by the equation,

$$value = \frac{\frac{Pixel\ grey\ level\ sum}{black\ rectangle}}{sum\ of\ black\ and\ white\ area} \tag{2.6}$$

The value was processed with the help of a cascaded classifier which is AdaBoost classifier as it is a highly accurate classifier. AdaBoost classifier is a combination of many inaccurate and weak classifiers which in term minimize the error rate (Micilotta et al., 2005). When the images are process by the AdaBoost classifier, the algorithm will select the most suitable weak classifier to determine it. If the weak classifier missed out some samples, the algorithm will select another classifier to identify the remnants samples. Since a 24 * 24 image contains million of Haar-like features, AdaBoost algorithm which acted as a learning algorithm to train the cascade classifier could provide a more effective method in human head detections. Furthermore, to improve the speed of recognition, region of interest (ROI) was introduced to the detection phrase as shown in Figure 2.12. By setting two virtual lines on the image file, the classifier would only need to detect the human head in between the lines. Hence, the response of the system would be much faster.

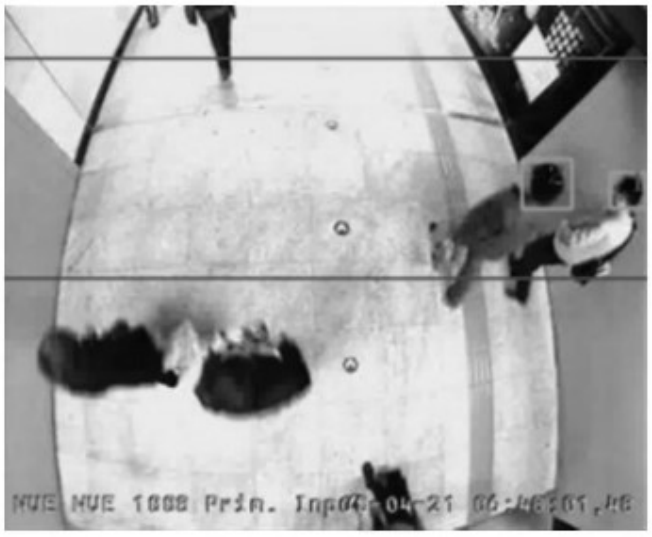

Figure 2.12: Recognizing pedestrian head in frame image (Chen et al., 2014)

2.2.6 HOG-PCA Extraction Method

Histogram of oriented gradients, also known as HoG proposed by Dalal et al. (2005) is still the most robust feature representation in pedestrian detection. The gradient of each pixel is processed and quantized into angular bins. The normalized quantized gradients by grouping pixels form the histogram of oriented gradients. A conclusion was made by Na (2012), in the paper it is stated that SVM + 500-dimensional HOG-PCA features had a higher detection accuracy compared to AdaBoost and Haar-like feature classifier. HOG-PCA consisted of two parts, HOG features and PCA feature. In HOD features, the events of gradient orientation were counted in a restricted portion of image. The technique was used for object detection. HOG descriptor would have described the appearances and the shape of the object by the distribution of edge direction or intensity gradients. Since it was not sensitive to light, the feature came with high computational cost (Qiang et al., 2006).

HOG features was extracted with 16x16 pixel block of four 8x8 cells from a 64x128 detection window and the support of 8x8 skip-step. Carry on, a 3780-dimensional HOG feature vector was extracted but not all dimensions were useful. Therefore, PCA technique was implemented to the system to reduce detection time and improving the accuracy of the classification at the same time.

PCA, principle component analysis was a famous feature extraction method as it was able to remove the second-order correlation of an ergodic process. PCA transform a high-dimensional input vector to the lower dimensional input vector where the components are unrelated with the help of the eigenvectors of covariance matrix (Du et al., 2006). As the transformation occurred, PCA method were able to expose the internal data structure. For better discriminating performance, PCA method was used to select the HOG feature vector with the highest informative dimensions as the final feature vector space.

By training the SVM classifier with HOG and HOF-PCA feature, their performances such as detection rate and classification time were compared. In the paper of Na (2012), a result was obtained of their comparison.

Table 2.1: Comparison Result of the Classifiers (Na et al., 2012)

Pedestrian Classifier & Features	Detection accuracy
Adaboost + 696-dimensional Haar-like features	95.1%
SVM-Adaboost + 100-HOG features	93.98%
SVM + 500-dimensional HOG-PCA features	96.83%

2.2.7 Shadow Removal Method

Shadow removal is one of the image processing method which was to increase the detection efficiency as the algorithm might mistakenly detected the shadow as the target (Zhao et al., 2011). Traditionally, image processing focused on analysing single pixel and ignored spatial structure relationship between each pixel as they were easily influence by the small changes such as external environment factor or camera noises. As a result, it would contain uncertainty on its luminance or colour. The objective of the shadow removal was to increase detection accuracy by avoiding getting wrong feature information and shadow of the target. In term of grayscale images, shadow and target normally had similar visual effect. To improve shadow detection result, colour information such as HSV and RGB colour space were added into the detection phrase.

HSV colour space reflected the colour better than the RGB colour space in term of human perspective. Although HSV colour space had the upper hand, in certain cases, the colour information of both the target and shadow will have same visual effect, RGB colour space would be a better choice in identifying whether the pixel belongs to the one or another. When all the information was dealt with RGB colour space, the first-order gradient images of 3 different channels as shown in Figure 2.13 will be obtained. Although there were three channels of information, they are quite similar to each other. Therefore, average of three channel for x and y

direction was considered instead of all three channels to reduce the computation complexity.

$$I_{grx(x,y)} = \frac{1}{3}\left(I_{rx(x,y)} + I_{gx(x,y)} + I_{bx(x,y)}\right) \quad (2.7)$$

$$I_{gry(x,y)} - \frac{1}{3}\left(I_{ry(x,y)} + I_{gy(x,y)} + I_{by(x,y)}\right) \quad (2.8)$$

Furthermore, two criterions were set to differentiate the shadow and targets. First criterion C0 was used to judge the pixel in term of HSV colour space information while another criterion which is C1 judged by using the RGB colour space information.

C0:

$$\frac{C_k^V(x,y)}{B_k^V(x,y)} < T^V \cup (C_k^S(x,y) - B_k^S(x,y) > T^S)$$

$$\cup (abs(C_k^H(x,y) - D_k^H(x,y) > T^{H1}) \cap (C_k^H(x,y) < T^{H2})) \quad (2.9)$$

C1:

$$(i_x(x,y) - B_x(x,y))^2 + (i_y(x,y) - B_y(x,y))^2 > 9\tilde{\sigma}^2 \quad (2.10)$$

$$\frac{C_k^V(x,y)}{B_k^V(x,y)} < T^V \quad (2.11)$$

$$C_k^S(x,y) - B_k^S(x,y) > T^S \quad (2.12)$$

$$C_k^H(x,y) < T^{H2} \quad (2.13)$$

Target or shadow that was protruded onto the background. As the process occurs, it will cause changes in brightness. While the changes on the saturation of pixel was large, the pixel was determined as target and vice versa for shadow.

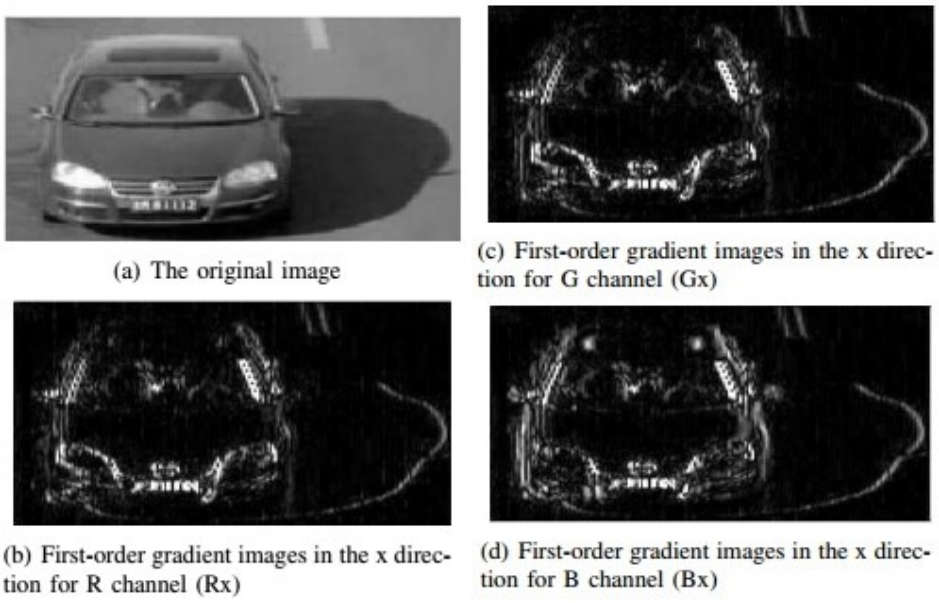

Figure 2.13: First-order gradient images in the x-direction for R, G, B channels (Zhang, 2011)

2.2.8 Optical Flow Method

Optical flow approach is the estimation of motion in video by matching points on objects over consecutive frames. It is used to describe the motion of the points or feature under constant spatial smoothness (Xiaofei et al., 2010). Optical flow based segmentation detects the moving regions with the flow vector characteristic of the moving object in a sequences of image over a certain period. The pro of the using optical flow method is that it is robust to multiple object motions, hence making it ideal for pedestrian tracking in crowd condition.

In addition, it can be used to detect moving object not only in the static position but also with camera motion. However, the con of the optical flow method is that it is vulnerable to image noise, colour and lighting condition as it required large computation requirement. Furthermore, it is very sensitive to motion discontinuities. For practical purpose, a specialized hardware would be required for accurate measurements due to the complexity of the algorithm and moderately high frame rate (Joshua et al., 2010).

Initially, background determination by the difference in optical flow for the different parts of a human body is used. However, due to the inability to detect static target, the method was dropped. Kalman filtering, condensation algorithm, mean shift and some other statistical method are becoming popular for pedestrian tracking nowadays. Kalman filtering classified the pedestrian movement with their current, previous position and time as parameters whereas mean shift reduce the changes between target and model histogram of the detected pedestrian (Dorin et al., 2000).

Due to the ability of estimating the state of process, it is able to predict the future states. It is the best filter among all other linear filter when involving Gaussian noise (Prasad 2012). Therefore, it is also implemented along with the Gaussian Mixture Model for background subtraction. From the paper of Vasileios et al. (2010), it is stated that the best way to compute object tracking is by combining both mean shift and Kalman Filtering method.

CHAPTER 3

METHODOLOGY

3.1 Introduction

This chapter will provide an overview of the detection system developed. To complete this project, a setup resembling a real surveillance system is built. The proposed and actual setup is discussed in this chapter. The positioning of the camera is one of the crucial elements in building a good detection system. Besides that, a stable and efficient algorithm is a must for the completion of this project. The reason for installing the camera overhead facing downwards is discussed. The main equipment used in this project is normal camera connected through the USB cables.

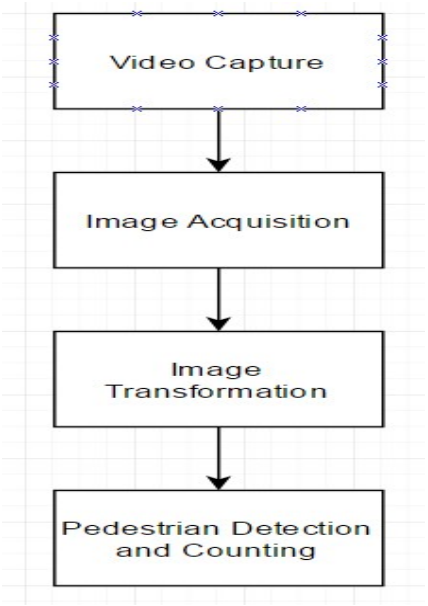

Figure 3.1: Flow chart of pedestrian counting system

3.2 System Setup

Figure 3.2 shows the proposed system setup for this project. An overhead camera was connected through a USB cable will be installed facing downwards or slightly slanted. If the camera is IP based, it will be connected to the switch module or a control unit before it was connected to the computer. The reason of the installation of camera and the camera type chosen will be discussed in the section below.

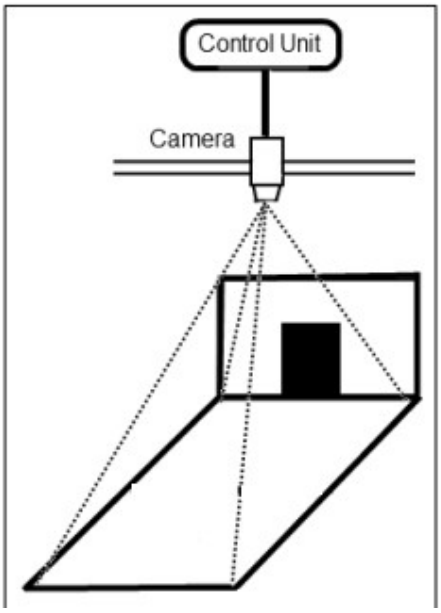

Figure 3.2: Proposed system setup

The height of the camera will affect the probability of the detection directly. The higher the camera height, the bigger the field of viewing of the camera and resulting a larger surveillance area. Although larger surveillance area could result in a better detection rate due to the pedestrian as they have more time to remain in the detection area. However, an appropriate height must be found as when the camera is too far away from the target, it loses focus and could not detect the desired target. On the other hand, if the camera is too near to the target, the detector might ignore the target as it is too big to be recognize as human. Furthermore, it might easily skip through the entire surveillance area easily

3.2.1 Problem faced by Camera Positioning

Normally, most of the video surveillance system is installed at an angle less than 45 degrees facing the surveillance area. For this type of installation, occlusion will happen which indicated the blocking of object with another object. This is a critical issue in implementing video analysis in video surveillance system. For example, when detecting the hallway or roadside, the system will not be able to detect multiple target as they are too many of them and blocking of the target will affect the detection accuracy.

In the paper of Chen (2006), a colour-based video camera is installed overhead at 4.2m above the ground to count the pedestrian passing through the detection area. The authors of the research had tested the system by using various moving patterns such as merging and splitting. By installing the camera at facing downward position, the system developed has an accuracy of 85% in counting number of pedestrian passing through the detected area.

In addition, a pedestrian detection system's research carried out by Bozzoli et al. (2007) which used a low cost camera on the ceiling on a public transport station had also installed its camera facing downward to the detection area. The data collected for number of people passing through the detected area had allowed the public transport governing bodies to optimize the route allocation reduce unnecessary costing and other services. The authors had mentioned that the developed system had an accuracy rate of around 95%.

Based on all these researchers, it can be deduced that, by installing the overhead camera while facing downward, the effectiveness of the detection system can be maximized. Therefore, this installation method is adopted in this project.

3.2.1 Camera choice

Camera can be separated into two types, fixed-camera or PTZ camera. Then, they can be further separate into two types, colour or non-colour. However, all types of camera can transmit data to the computer and or control unit. For this project, a

commercial low-cost camera was used. The camera used has a closest focusing length of 4.8mm. It contains a high-quality CMOS sensor, high frame rate (up to 30fps) video playback and minimum sensitivity of 2.0V/Lux.Sec. Besides that, it supports high-quality AVI, MPEG video recording while image storing of BMP, JPG format function. It output frame is 640x480 resolution which is very suitable for the detection system.

However, if possible, an IP network camera is a better choice in video analytisis. IP camera is preferred due to its flexibility to stay connected far away from the controlling unit through LAN cables. If compared to the USB port camera, it has a higher transfer rate and a stable data transmission. Furthermore, if more than one cameras are being used, a switch module could be used as the IP network camera is recognized by IP address as their device unit number. IP camera usually has various type of video output such as H.264, MPEG-4 or Motion JPEG (MJPEG).

The video format used in this project is AVI file format which stands for Audio Video Interleaved. The crucial reason of choosing this video format is because it is supported in any kind of platforms and software. Furthermore, there are various types of codec available which can be used to create a smaller storing size version of compressed AVI format file. It is developed by windows, so it is compatible for Microsoft operating systems. In addition, it did not require any specific hardware or supportive software acceleration to run the video or audio file. This will reduce the burden or memory space needed for control unit by reducing the computation burden.

3.3 Image Processing Library

For this project, Microsoft Visual Studio along with Open Source Computer Vision Library (OpenCV) is used for developing the pedestrian counting system. As the advancement of technology, OpenCV is enough to support the whole detection algorithm. Within the default libraries in the OpenCV, there are various function to process the video feed from the various sources.

3.3.1 Image Morphology

In order to load an image, the function imread loads an image from a source and returns it. If the image could not be load due to missing, corrupted or improper permissions and invalid format, the functions will return an empty matrix and close the application.

Furthermore, fuction imwrite the opposite of the imread. Imwrite is used to save the output image according to the desired format and location.

For video capturing from the video files, image sequences or cameras, the function VideoCapture is provided in the OpenCV library.

An example code is shown below.

```
#include "opencv2/opencv.hpp"

using namespace cv;

int main(int, char**)
{
    VideoCapture cap(0); // open the default camera
    if(!cap.isOpened())  // check if we succeeded
        return -1;

    Mat edges;
    namedWindow("edges",1);
    for(;;)
    {
        Mat frame;
        cap >> frame; // get a new frame from camera
        cvtColor(frame, edges, CV_BGR2GRAY);
        GaussianBlur(edges, edges, Size(7,7), 1.5, 1.5);
        Canny(edges, edges, 0, 30, 3);
        imshow("edges", edges);
        if(waitKey(30) >= 0) break;
    }
    // the camera will be deinitialized automatically in VideoCapture destructor
    return 0;
}
```

Figure 3.3: Example OpenCV code

3.3.2 Geometric and Miscellaneous Image Transformations

Resizing is the most frequently used operation in image processing. OpenCV offers a function called resize where the input image is scale down or up to a specified size.

However, to shrink or enlarge an image, it is better to use the function CV_INTER_AREA. CV_INTER_CUBIC and CV_INTER_LINEAR. The image capture is resized to 640x480, this is the optimum size for prebuilt HOGDescriptor in OpenCV library.

Furthermore, function cvtColor is also used together in image transformation. The obtained frames from the video capture or video feed is converted to the grayscale. RGB to grayscale conversion is a very simple process and it was done by most of the detecting system. In addition, which is also the main reason for conversion, it is fast. When the detector acts on the grayscale image, the processing speed is boosted and can be easily applied on a real time basis. Furthermore, by converting the image to grayscale, a gradient for the image is obtained by (R+G+B)/3. This give us a faster computation speed. As mentioned in Chapter 2, R channel, G channel and B channel have almost same pixel values. Therefore, when we computed the gradient of the image, we can save out time by a quite big factor.

3.4 Pedestrian Counting

The pedestrian detection algorithm is based on the Dalal-Triggs Algorithm which has been implement into OpenCV library. Each detection window is divided into cells of size 8x8 pixels and a sliding window of 16x24. This will only detect people at a single scale. However, to achieve multiple detection, different threshold was tried for different range of camera viewing angle and distance from the desired detection area. The scaling window is also increment by the multiple of block stride. It is important to redefine that no motion or tracking algorithm are used in the system. Brute force searching over the captured frame over the time and the target was detected by using HOGDescriptor::detectMultiScale function.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter focuses on discussion and results obtained from the pedestrian counting system developed in Microsoft Visual Studio and OpenCV. Histogram of Gradient (HOG) and Support Vector Machine (SVM) approaches are the most popular and successful detector in the word. HOG is chosen over SVM due to its simplicity to be understand in compared to other object recognition method. Furthermore, another reason for using HOG is that it globalized the feature instead of storing feature locally. This means that, for HOG, an entire person is represented by a single feature vector. When compared to other method, each and every parts of a human body will be represented by different feature vectors.

4.2 Parameter of Algorithm

In this project, different parameters of the Histogram of Gradient (HOG) algorithm had been discussed. As discussed in the previous chapter, conventional view is different from the overhead view. Hence, the resulting data obtained is also different with each other. So, different parameters are needed to be tuned before applying HOG. The default HOG parameters by Dalal-Triggs was used to detect the pedestrian on the road side, the parameter which are win_stride, padding, group threshold, block size, cell size and so on. It is as table below:

Table 4.1: Different HOG parameters

Test No.	Win_stride	Padding	Scaling factor	Group threshold	Blocking Overlapping	Accuracy, %

Default	8x8	16x16	1.08	2	No	50.0
1	8x8	0x0	1.08	2	No	50.0
2	8x8	0x0	1.08	0	Yes	150.0

Group threshold is a coefficient to regulate the similarity threshold, it determines when should the detected parts to place in the group. Normally, when detected, some object could be covered by many rectangles as shown in the Figure.

Figure 4.1: Multiple rectangle on a single target

When this happen, the counting of pedestrian tends to go wrong. This is because the pedestrian counter is normally based on the HOG detected. As number of rectangle increased, the detected number of people also increased. Therefore, by assignment value of 2 to the grouping threshold, it acts as a final threshold which remove or joint all the existing nearest rectangle to be 1. Low value of group threshold results zero grouping while higher value provides result grouping. Therefore, an optimum threshold value must be obtained through trial and error.

Whereas for padding, it adds a certain number of extra pixels the both side of the input image. This will make the sliding window or detection window to be placed a bit outside than the original dimension of the input image. This is used to detect the

people who are at the edge of the input image. When there is no one at the edge of the image, the presence of padding becomes irrelevant. Therefore, it has the same detection accuracy.

Figure 4.2: Detection on 16x16 padding and 0x0 padding

From the same table again, we can see that, there is no different between two set of padding value. Each of them has detected 4 out of 8 persons in the figure. However, when the input frames were fetched from the overhead camera, the different between these size of padding are shown.

Figure 4.3: Detection on 0x0 block size during run time

The third parameters would be win-stripe. Win-stripe is like a brain that determines and tells the HOG to move the detection window a certain amount by sliding from the left to the right and repeated until the end of the image. A small win-stride will provide higher accuracy while on the other hand, a larger win-stride decrease the computational time. Therefore, this value will also require tuning and testing to get the optimum value for the best performance.

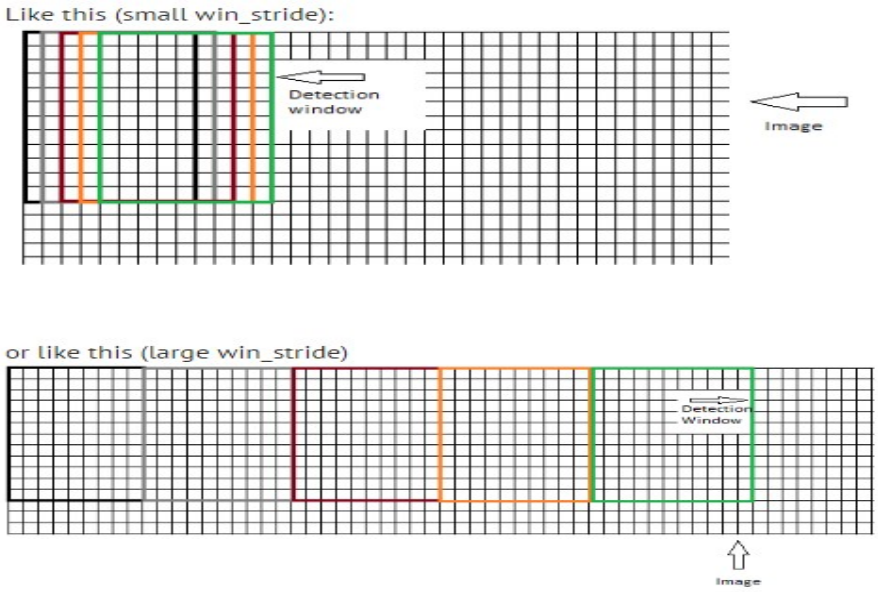

Figure 4.4: Win_stride

The last parameter would be the scaling factor. The higher the scaling factor, the higher the coefficient of the detection window increase. Therefore, when the scaling factor is more than 1.5, the frame rate of the output window increases. However, when the frame rate increases, the time for detector to detect the current frames decrease. Therefore, the accuracy of the system decreases. After trial and error, the error, 1.08, this is the optimum value for the developed detecting system.

4.3 Recorded Videos

In order to verify the performance of the system, the video feed from the camera on the surveillance area are recorded. The video recorded is in AVI format with sufficient lighting condition with an overhead camera.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

From this project, a pedestrian counting system capable of detecting and counting people was successfully developed. This project utilized a commercialized low cost overhead camera with open source software and a control unit such as CPU.

The implementation of the pedestrian counting system could be used on embedded unit such as microprocessor and Raspberry Pie. The advantages of doing it is due to the affordability of the system. An embedded unit at its maximum price is also much cheaper than a complete CPU. This in turn reduces the start-up cost for the pedestrian counting system. Besides that, embedded system could also be implemented along with other semiconductor products and resulting a more complete software instead of controlling by using CPU. The main advantage is that the embedded unit is closely related to the semiconductor products. Therefore, they can easily join together and perform some unexpected outcome. Furthermore, embedded system is less likely prone to failure.

5.2 Problem faced and Further Improvement

In this section, problem faced and further improvement have been discussed. The pedestrian counting system developed is simple but at the same time it has many disadvantages such as frame rate, detector scaling, accuracy and even counter.

For the algorithm developed in this project, the response or the frame rate of the pedestrian detector is very low. The main reason for this is the HOG descriptor used. Although the default library of the HOG descriptor provided by OpenCV is completed, the efficiency of the system is very low. For example, the provided library only works on the multiplication of 640 and 480. Any scaling outside this

region will affect the stability of the system. Therefore, it is wise to train our own HOG descriptor instead of using the provided library.

In the paper of Dalal (2005), the authors also had shown us that, a good way of doing a pedestrian counting system was to developed our own descriptor. In their research, a pedestrian database containing 1239 of images as positive training samples and 1239 negative samples. Then the training image sets were fed into the system and generated the detectors. Then, an initial detector is trained with different parameter combination. After that, the preliminary detector was re-trained with another supplement set of parameter combination to produce the final version of detector. As a result, the final descriptor contained approximately 1.7Gb of RAM for SVM training. This kind of descriptor would have a very low error and the parameters of the detector could be tuned more effectively to achieve a better detection rate.

Another problem faced is the counting. In this project, the pedestrian was counted in the current frame when it is being detected. To solve this, region of interest (ROI) method should be implemented. However along with this method, tracking system such as background subtraction, optical flow and so on will be the add-on for this system. In the paper of ZhaoXiang (2011), the authors had stated that the function cvCalcOpticalFlowPyrLK implemented in OpenCV is incomplete. Although the function is able to calculate optical flow by using iterative Lucas-Kanade (LK) method, the optical flow of the very first frame had some problem. As the calculation of optical flow on N frame, the N-1 frame is needed. Therefore, for the first frame of optical flow method, the function has no way to bootstrap itself.

Lastly, the discussed problem faced is the system limitation. The system is unable to differentiate between two persons if they are walking side by side or stick closely to each other when passing through the surveillance area. Furthermore, the range of detection is also limited by hardware components. A high focus length camera is more preferred as they have a large viewing distance. Hence providing a clearer and sharper view inside the detection zone. Due to the limitation of the camera, the video feed is not in good condition. When being passes through the HOG

detector, some target couldn’t be located due to the shaded area cause by the tree is too deep in colour and affect decision making of the detector.

REFERENCES

Candamo, J., Shreve, M., Goldgof, D., Sapper, D. and Kasturi, R. (2010). Understanding Transit Scenes: A Survey on Human Behavior-Recognition Algorithms. *IEEE Trans. Intell. Transport. Syst.*, 11(1), pp.206-224.

Cetinkaya, H. and Akcay, M. (2015). People Counting at Campuses. *Procedia - Social and Behavioral Sciences*, 182, pp.732-736.

Chen, Y., Guo, S., Zhang, B. and Du, K. (2013). A Pedestrian Detection and Tracking System Based on Video Processing Technology. *2013 Fourth Global Congress on Intelligent Systems*.

Chinchkhede, D. and Uke, N. (2012). Image Segmentation in Video Sequences using Modified Background Subtraction. *International Journal of Computer Science & Information Technology*.

Comaniciu, D., Ramesh, V. and Meer, P. (2000). Real-time tracking of non-rigid objects using mean shift. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No.PR00662)*.

Conrad, G. and Johnsonbaugh, R. (1994). A real-time people counter. *Proceedings of the 1994 ACM symposium on Applied computing - SAC '94*.

Dalal, N. and Triggs, B. (n.d.). Histograms of Oriented Gradients for Human Detection. *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

Du, K. and Swamy, M. (2006). *Neural networks in a softcomputing framework*. London: Springer.

Elgammal, A., Harwood, D. and Davis, L. (2000). Non-parametric Model for Background Subtraction. *Lecture Notes in Computer Science*, pp.751-767.

Green-Roesel, R., Diógenes, M., Ragland, D. and Lindau, L. (2008). Effectiveness of a Commercially Available Automated Pedestrian Counting Device in Urban Environments: Comparison with Manual Counts.

Haritaoglu, I. and Flickner, M. (n.d.). Detection and tracking of shopping groups in stores. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*.

Hashimoto, K., Morinaka, K., Yoshiike, N., Kawaguchi, C. and Matsueda, S. (n.d.). People count system using multi-sensing application. *Proceedings of International Solid State Sensors and Actuators Conference (Transducers '97)*.

Ji, X. and Liu, H. (2010). Advances in View-Invariant Human Motion Analysis: A Review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1), pp.13-24.

Kamel, M., Fkry, M., Mashat, A., Biqami, N., Barhamtoshy, H. and Bedewy, I. (2004). Monitoring, Surveillance and Control of the Crowds in the Holy Sites Using SCADA System.

Keitaro, K. (2012). A Framework of Vision-Based Detection-Tracking Surveillance Systems for Counting Vehicles.

LEFLOCH, D. (2007). Real-Time People Counting system using Video Camera.

Micilotta, A., Ong, E. and Bowden, R. (2005). Detection and Tracking of Humans by Probabilistic Body Part Assembly. *Procedings of the British Machine Vision Conference 2005*.

Oliver, N., Rosario, B. and Pentland, A. (2000). A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8), pp.831-843.

Paul, M., Haque, S. and Chakraborty, S. (2013). Human detection in surveillance videos and its applications - a review. *EURASIP Journal on Advances in Signal Processing*, 2013(1), p.176.

Prabhu, G. (2011). Automated Detection and Counting of Pedestrians on an Urban Roadside.

Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Avidan, S. (n.d.). Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*.

Shou, N., Peng, H., Wang, H., Meng, L. and Du, K. (2012). An ROIs based pedestrian detection system for single images. *2012 5th International Congress on Image and Signal Processing*.

Terada, K., Yoshida, D., Oe, S. and Yamaguchi, J. (n.d.). A method of counting the passing people by using the stereo images. *Proceedings 1999 International Conference on Image Processing (Cat. 99CH36348)*.

Velipasalar, S., Tian, Y. and Hampapur, A. (2006). Automatic Counting of Interacting People by using a Single Uncalibrated Camera. *2006 IEEE International Conference on Multimedia and Expo*.

Wang, X., Sun, J. and Peng, H. (2013). Foreground Object Detecting Algorithm based on Mixture of Gaussian and Kalman Filter in Video Surveillance. *JCP*, 8(3).

Yan-yan, C., Ning, C., Yu-yang, Z., Ke-han, W. and Wei-wei, Z. (2014). Pedestrian Detection and Tracking for Counting Applications in Metro Station. *Discrete Dynamics in Nature and Society*, 2014, pp.1-11.

Zhang, Z., Hou, Y., Wang, Y. and Qin, J. (2011). A traffic flow detection system combining optical flow and shadow removal. *2011 Third Chinese Conference on Intelligent Visual Surveillance*.

Zhao, X., Dellandrea, E. and Chen, L. (2009). A People Counting System based on Face Detection and Tracking in a Video. *Advanced Video and Signal Based Surveillance*, pp.67-72.

APPENDICES

APPENDIX A: Pedestrian Counting System Code

```
#include <opencv2/core/core.hpp>                //to ensure liability
#include <opencv\cv.h>                                      //to ensure
liability
#include "opencv2\opencv.hpp"                          //to ensure liability
#include <opencv2/highgui/highgui.hpp>          //image and window
#include <opencv\highgui.h>
#include <iostream>                        //to use cout
#include <string>
#include "opencv2\imgproc\imgproc.hpp"

using namespace cv;
using namespace std;

String inttostr(int input)
{
       stringstream ss;
       ss << input;
       return ss.str();
}


bool activation = true;
bool pause = false;
Mat frame, frame2, grayframe, grayframe2, differenceImage, thresholdImage,
outframe;
String imagename;
int x = 0, y = 0, key1 = 0x0;
int people_count = 0;

int main(int argc, char* argv[])
{

//     VideoCapture capture(0);

       VideoCapture capture("C:\\Users\\aeolu\\Desktop\\test.avi");

       if (!capture.isOpened())          //checking activation of video cam
       {
              cout << "Cannot open the video file !" << endl;
              return -1;
       }
```

```
	capture.set(CAP_PROP_FRAME_WIDTH, 640);
	capture.set(CAP_PROP_FRAME_HEIGHT, 480);
	namedWindow("video capture", WINDOW_AUTOSIZE);

	int frame_width = capture.get(CV_CAP_PROP_FRAME_WIDTH);
	int frame_height = capture.get(CV_CAP_PROP_FRAME_HEIGHT);
	VideoWriter video("C:\\Users\\aeolu\\Documents\\Visual Studio
2015\\Projects\\TR Copy\\Trial\\test"+inttostr(y)+".avi", CV_FOURCC('M', 'J',
'P', 'G'), 10, Size(frame_width, frame_height), true);




	HOGDescriptor hog;
	hog.setSVMDetector(HOGDescriptor::getDefaultPeopleDetector());
	//hog.setSVMDetector(HOGDescriptor::getDaimlerPeopleDetector());

	while (activation)
	{
//		outframe = imread("C:\\Users\\aeolu\\Desktop\\Testing1.jpg",
CV_LOAD_IMAGE_GRAYSCALE);
		capture.read(frame);

		video.write(frame);
//		imshow("Example1", frame);		//show frame in the
Example1

		resize(frame, outframe, Size(640, 480), 0, 0, INTER_CUBIC);
//		resize(frame, outframe, Size(480, 720), 0, 0, INTER_CUBIC);
		cvtColor(outframe, outframe, CV_BGR2GRAY);

		//In-space filtering
		GaussianBlur(outframe, outframe, cv::Size(5, 5), 0);

		if (outframe.empty())
	//checking frame conversion
		{
			cout << "Cannot read a frame from video stream" << endl;
			break;
		}

		if ((key1 = waitKey(30)))
		{
			switch (key1)
			{
			case 27:			//'ESC'
//				cvReleaseData;
//				cvReleaseVideoWriter;
				activation = false;
				break;

			case 32:			//'Space'
				imagename = "C:\\Users\\aeolu\\Documents\\Visual
Studio 2015\\Projects\\TR Copy\\Trial\\Testing" + inttostr(x) + ".jpg";
				imwrite(imagename, frame);
				x++;
				break;
			}
		}

		vector<Rect> found, found_filtered;
```

```
            hog.detectMultiScale(outframe, found, 0.0, Size(8, 8), Size(32,
32), 1.05, 2);
//          hog.detectMultiScale(outframe, found, 0.0, Size(8, 8), Size(32,
64), 1.2, 1);
            size_t i, j;

            //filter redundant rectanalges
            for (i = 0; i < found.size(); i++)
            {
                  Rect r = found[i];
                  for (j = 0; j < found.size(); j++)
                        if (j != i && (r & found[j]) == r)
                              break;
                  if (j == found.size())
                        found_filtered.push_back(r);
            }

            for (i = 0; i < found_filtered.size(); i++)
            {
                  Rect r = found_filtered[i];
                  // the HOG detector returns slightly larger rectangles
than the real objects.
                 // so slightly shrink the rectangles to get a nicer output.
                  r.x += cvRound(r.width*0.1);
                  r.width = cvRound(r.width*0.6);
                  r.y += cvRound(r.height*0.07);
                  r.height = cvRound(r.height*0.8);
                  rectangle(outframe, r.tl(), r.br(), Scalar(0, 255, 0), 3);
            }
            //pedestrian counting
            people_count = found_filtered.size();



            int fontFace = FONT_HERSHEY_COMPLEX_SMALL;
            double fontScale = 1;
            int thickness = 1;
            Point textOrg(0, 460);
            string Text = "Detected Pedestrian No.: "+
inttostr(people_count);
//          putText(outframe, Text, textOrg, fontFace, fontScale,
Scalar::all(255), thickness, 8);
            putText(outframe, Text, textOrg, fontFace, fontScale,
Scalar(CV_FONT_BLACK), thickness, 8);

            imshow("video capture", outframe);
            cout << "Total Detected number of people in current frame: " <<
people_count << endl;
      }
      //release the capture before re-opening and looping again.
//    capture.release();
      return 0;
}
```