

School of Public Administration  
Bachelor of Science in Computing

**COMP491 Final Year Project  
Progress Report**Academic Year 2018/19

|  |  |
| --- | --- |
| Pattern Recognition using Machine Learning | |
|  |  |
| Project number: | 20 |
| Student ID: | P-15-0792-1 |
| Student Name: | Zhiming Lin, Tony |
|  |  |
| Supervisor: | Dr. Yue Liu, June |
| Assessor: | Dr. Yang Xu |
|  |  |
| Submission Date: | 15/11/2018 |

Declaration of Originality

I, Zhiming Lin, declare that this report and the work reported herein was composed by and originated entirely from me. This report has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given in the bibliography.



11/11/2018

Abstract

Nowadays, traffic congestion is a serious problem in every city. To relieve this situation, all sorts of traffic control techniques are applied. The main goal of this project is to use images from Transport Bureau of the Macao S.A.R. website (DSAT) [1] to monitor the traffic and predict the traffic congestion via detecting vehicles on the roads of Macao. Several algorithms will be investigated and implemented in this project under the local circumstance with the consideration of the limitations of DSAT surveillance materials. One is based on histogram of oriented gradients (HOG) with support vector machine (SVM) [5]; another one is YOLO [7] which is a well performance detection model.

There are also some projects using object detection algorithms to detect cars on the roads. However, unlike them, this project is using up-to-date object detection algorithm with decent traditional object detection algorithm as a benchmark to detect vehicles on the roads especially for Macao. This ensures that the project is best fit for Macao traffic situation.

Table of Contents

[1 Introduction 7](#_Toc530144408)

[1.1 Objectives 7](#_Toc530144409)

[1.2 Risk Assessment 8](#_Toc530144410)

[1.3 Summary 9](#_Toc530144411)

[2 Background and Related Work 10](#_Toc530144412)

[2.1 TensorFlow [2] 10](#_Toc530144413)

[2.3 Scikit-learn [3] 10](#_Toc530144414)

[2.4 Keras [4] 10](#_Toc530144415)

[2.5 Object Detection 10](#_Toc530144416)

[2.5.1 Histogram of Oriented Gradients with Support Vector Machine. [5] 10](#_Toc530144417)

[2.5.2 Convolutional Neural Network [6] 11](#_Toc530144418)

[2.5.3 YOLO [7] 11](#_Toc530144419)

[2.6 Related Work 12](#_Toc530144420)

[2.6.1 Traffic Analysis Method Proposed by A. Atvar, Y. Artan and Ş. Öztürk [8] ……………………………………………………………………………...12](#_Toc530144421)

[2.6.2 Traffic Congestion Analysis Proposed by J. Wan, Y. Yuan and Q. Wang [16] ……………………………………………………………………………...12](#_Toc530144422)

[3 Completed Work 13](#_Toc530144423)

[3.1 Preparation Work 13](#_Toc530144424)

[3.2 Histogram of Oriented Gradients with Support Vector Machine 16](#_Toc530144425)

[3.3 YOLO 21](#_Toc530144426)

[4 On-going and Future Work 24](#_Toc530144427)

[4.1 HOG+SVM 24](#_Toc530144428)

[4.2 YOLO 24](#_Toc530144429)

[5 Conclusion 25](#_Toc530144430)

[References 26](#_Toc530144431)

Table of Figures

[Figure 1: Probability impact matrix before proposed solution 8](#_Toc454957831)

Figure 2: YOLO architecture 11

Figure 3: Proposed procedure 12

Figure 4.1: Image from DSAT at 19:40 14

Figure 4.2: Image from Calgary at 19:40 14

Figure 5: One example of deleted image 15

Figure 6: One example of labelled image 15

Figure 7: Car dataset 16

Figure 8: Non-car dataset 17

Figure 9: Linear SVM 17

Figure 10: Flowchart of HOG+SVM 18

Figure 11.1: Original image 18

Figure 11.2: HOG of the image 18

[Figure 12: Sliding window 19](#_Toc454957831)

[Figure 13.1: Test Image 1 using HOG+SVM 20](#_Toc454957831)

[Figure 13.2: Test Image 2 using HOG+SVM 20](#_Toc454957831)

[Figure 14.1: Test Image 1 using HOG+SVM with NMS 20](#_Toc454957831)

[Figure 14.2: Test Image 2 using HOG+SVM with NMS 20](#_Toc454957831)

[Figure 15: Test Image 1 using YOLO 22](#_Toc454957831)

[Figure 16: Test Image 2 using YOLO 22](#_Toc454957831)

[Figure 17: YOLO in clearer image 23](#_Toc454957831)

Figure 18.1: Gantt chart 1 26

Figure 18.2: Gantt chart 2 26

List of Tables

[Table 1: Table of prioritized risk 1](#_Toc449539554)

# Introduction

Nowadays, the traffic congestion is one of the major problems in metropolitan cities, such as Macao. Many people complain about the time they have to spend on their commuting especially during the rush hours. Thankfully, the Transport Bureau of the Macao S.A.R. website (DSAT) [1] provides online real-time images from traffic cameras in main roads of Macao. Thus, people can have general ideas about current traffic congestion level and avoid traffic jams in certain roads. However, it does not provide the exact number of vehicles on the roads and people have to subjectively predict the congestion by counting the cars by themselves which is not convenient and also inaccurate at all. It would be better that the real-time object detection is implemented to detect the volume of cars on the particular roads. Therefore, the project is designed to solve this problem.

It is quite well known that object detection has been discussed over a decade and many great algorithms and ideas have been introduced to improve the accuracy of object detection. However, there are only a few implements are specially designed for images from traffic cameras. Also, no literature has shown which algorithm can fit into Macao local case as the traffic images obtained from DSAT website has lots of limitations compared with traffic images from other cities. Because the roads in Macao are quite narrow and the images from DSAT are not clear but blurred. To find out the one which can be applied to Macao situation with better performance or improved performance is the highlight of this project.

## Objectives

The goal of this project is to design and implement an object detection algorithm for detecting the vehicles from the images of DSAT. It would help people to find out the congestion level on the roads. Nowadays, there are many image recognition and machine learning algorithms to detect objects. However, not all the algorithms are suitable for recognising cars from traffic cameras. Therefore, literatures review on object detection using machine learning especially for vehicles should be done to conduct better results. It is also necessary to do some analysis based on the detection results.

The objectives of this project are:

* Identify the problem of image recognition: traffic pattern recognition.
* Do literatures review on image recognition and machine learning algorithms.
* Implement a machine-learning algorithm to recognise the traffic condition.
* Test the algorithm using real-time traffic image crawled from Transport Bureau of the Macao S.A.R website (DSAT) [1] website and analyse the traffic results.
* Compose a final project report.

## Risk Assessment

Risk one: Training data may be lost.

Solution: It is important to have back-up training data because this is crucial to whole project. Thus, the data should have at least one copy in different places such as USB (Universal Serial Bus) flash drive or cloud drive. Once the original data is lost, the backup data can be used immediately.

Risk two: Images quality may be too low to train.

Solution: The image quality is another significant point which affects the quality of the training data and the accuracy of the training results. In order to prevent such things happening, images which are from DSAT website when the weather is or the sky is dark should not be used. Instead, the images which have good lighting condition should be used for training.

Risk three: Crawling images from DSAT may have different sizes or types of images. Solution: Images size and type are two critical parameters which should be unified before training them. Therefore, before training data, images from DSAT, should be classified in terms of place and unified in same size and type.

Table 1: Table of prioritized risk

|  |  |
| --- | --- |
| Priority | Risk Identifier and Description |
| 1 | Risk 1: Images quality may be too low to train. |
| 2 | Risk 2: Crawling images from DSAT may have different sizes or types of images. |
| 3 | Risk 3: Training data may be lost. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Probability** | High |  |  | Risk 1 |
| Medium |  | Risk 3 | Risk 2 |
| Low |  |  |  |
|  |  | Low | Medium | High |
|  |  | **Impact** | | |

Figure 1: Probability impact matrix before proposed solution

## Summary

Aim at using machine learning to detect objects in the real-time online DSAT website and further discuss about the results, this report is organised as follows: Chapter 2 introduces the background and related work. Chapter 3 presents the work has completed. Chapter 4 shows the future work of this project. In chapter 5, the conclusion is given of this project.

# Background and Related Work

## TensorFlow [2]

TensorFlow is the tool which is going to be used in this project since it provides several tools that can ease the development. This tool is developed by Google which is the perfect tool for building deep learning network. Also, TensorFlow is allowed to deploy different platforms and across different operating systems. This powerful and open source software can speed up the development time.

## Scikit-learn [3]

Scikit-learn is an open-source machine learning library of providing useful algorithms and functions to develop this project. Furthermore, it provides detailed documentation which can only provide solutions to solve problems but also give more information to tune the model. The scikit-learn supports more than one machine learning model for users to implement and best fit situations. Thus, it is mainly used to train support vector machine which is one of the machine learning algorithms the project will implement.

## Keras [4]

Keras which is one of the Python deep learning libraries built on TensorFlow. Because of providing high-level neural networks API, it can help to reduce the work of building TensorFlow applications. Moreover, it supports Windows, Mac OS and Linux operating system which can lessen the difficulties of development in different platforms. Also, the nice documentation from Keras official website could ease the building machine learning model and speed up the developments.

## Object Detection

### Histogram of Oriented Gradients with Support Vector Machine. [5]

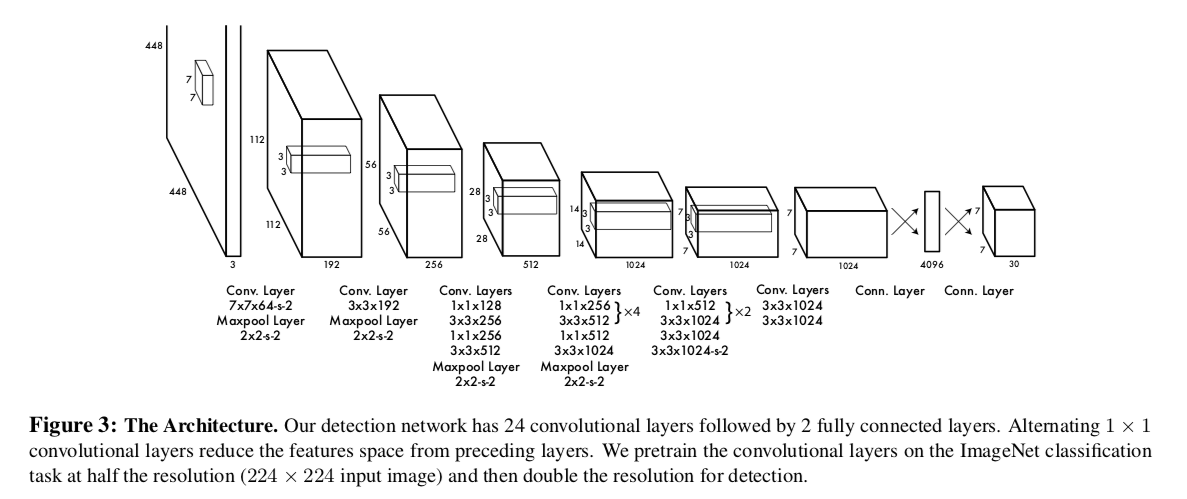
Before convolutional neural network [6] became the state-of-art machine learning models, histogram of orients gradients with support vector machine (HOG+SVM) was the most robust object detection method. This model would extract the histogram of oriented gradients from every training image, then it would use these histograms to train support vector machine to detect objects.

### Convolutional Neural Network [6]

Recently, the convolutional neural network has been improved the accuracy of object detection increasingly. This model allows people to work on deep learning to detect more complicated and difficult objects. The common convolutional neural network contains several layers and functions to better extract attributes of objects such as convolution layer, pooling layer and full connection layer.

### YOLO [7]

YOLO, you only look once, which is the state-of-art real-time object detection system. It is based on standard convolutional neural network with better performance. Unlike other detection systems, the YOLO only uses a single neural network to detect objects on the full image. This architecture makes the YOLO detects objects with extremely low latency. The latest version of YOLO is v3. Compared to v2 version, this v3 has better performance and better backbone classifier. The architecture is shown below (Figure 2).



**Figure 2 YOLO architecture [7]**

## Related Work

Nowadays, more and more people are coming travel or work Macao, and it is inevitable that this situation could worsen the traffic congestion in Macao. People are more likely to avoid wasting time on commuting.

### Traffic Analysis Method Proposed by A. Atvar, Y. Artan and Ş. Öztürk [8]

As this paper stated [8], the traffic camera is used to detect traffic intensity. A specific region is selected manually from the images of traffic cameras. In this paper, Leunger-Malik (LM) filter is used to extract attributes from the images. After that, the random forest algorithm which is one of the machine learning algorithms is used to determine the road occupancy. However, this may not apply to Macao situation perfectly because the roads in Macao are quite narrow and the congestion level is better defined by machine automatically. Also, the modern machine learning algorithms should be used to detect objects.

### Traffic Congestion Analysis Proposed by J. Wan, Y. Yuan and Q. Wang [16]

In this paper, new system is proposed to detect traffic congestion level. The image from traffic cameras are used to perform Inverse Perspective Mapping (IPM) into desired transformed image. After that, the transformed image is split into several regions to operate feature extraction. These features are merged together into final features which will be used to determine congestion level. The procedure can be seen in figure 3.

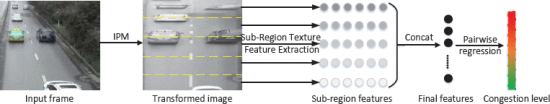


Figure 3 Proposed procedure [16]

Also, two Deep learning methods are used as comparisons. However, the results are not satisfied due to weak label of the vehicles’ positions and not well-tuned networks. Thus, the well-designed machine learning algorithms should be used to determine the congestion level.

# Completed Work

In this chapter, the way that implementation of object detection algorithms will be given and difficulties and how to tackle them will also be presented.

## Preparation Work

Before working on the project, it is better to do some research about existing works which are related to this project. Unfortunately, there are only few papers which are about solving traffic congestion using machine learning or deep learning through low-quality images from traffic cameras.

The quality of training dataset could bring huge difference in the results. Basically, the images will be segmented further to extract the attributes during the training. If the dataset has low resolutions or bad lighting conditions, it would be difficult for training models to extract and learn the attributes in the images.

To train the detection model, a dataset consisting of a large amount of labelled traffic images is required. There are two approaches can be used to solve this problem. Google Dataset Search [9] provides all sorts of dataset which also includes datasets usable for this project.

However, there are no such datasets with labelling of number of vehicles on the road from traffic cameras from the traffic cameras.

For example, there is a dataset of unlabelled images from traffic cameras provided by the city of Calgary, Canada [17]. This dataset is quite similar to images from DSAT but has significant differences between them. Firstly of all, the quality of images provided by Calgary has better resolutions with 840 pixels by 630 pixels compared to 352 pixels by 288 pixels from DSAT. Also, even the lighting condition is both bad in both images, the images from Calgary are much clearer than that from DSAT (Figure 4.1 and Figure 4.2). Secondly, the images from Calgary would be updated every minute, while the images from DSAT would be updated every 5 seconds. Also, the roads of Calgary are much wider than that of Macao with fewer vehicles which are crowded closely.

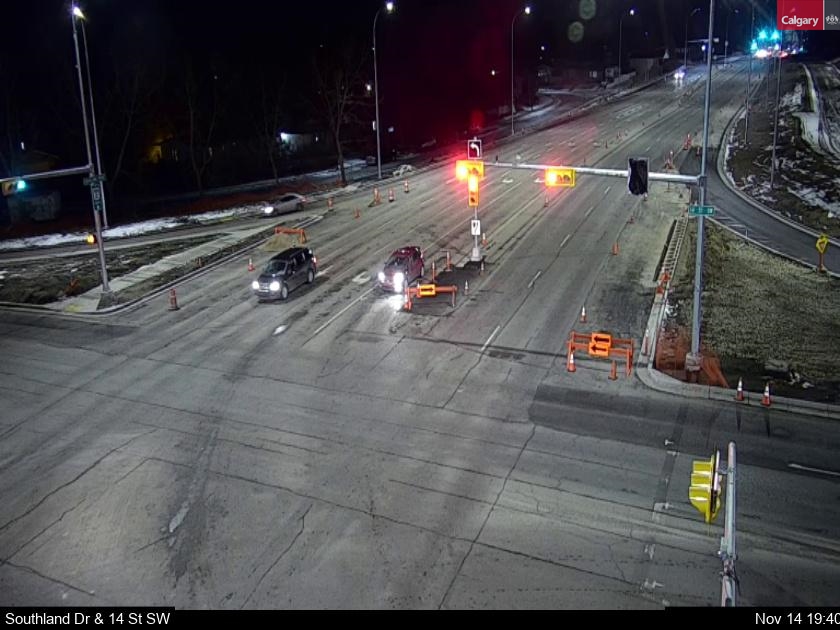


Figure 4.1 Image from DSAT at 19:40

Figure 4.2 Image from Calgary at 19:40

Therefore, on the one hand, training from existing segmentation of vehicles is worth trying. On another hand, the web crawling to crawl images from DSAT and process the images to make a new dataset especially from Macao traffic situation.

Currently, more than 60 thousand images from DSAT have been downloaded, but there are several issues when preparing the dataset. Firstly, images from DSAT should be cleaned. Some images should be deleted when there are no vehicles on the roads (Figure 5). Secondly, it seems that the server of DSAT would be stuck somehow and there is more than one image which is actually the same. Thirdly, when it comes to train this dataset, the images should be well labelled and about 300 images were labelled by using LabelImg [10] (Figure 6). The locations and areas of each vehicle appeared on the images are precisely labelled with bounding boxes.



Figure 5 One example of deleted image

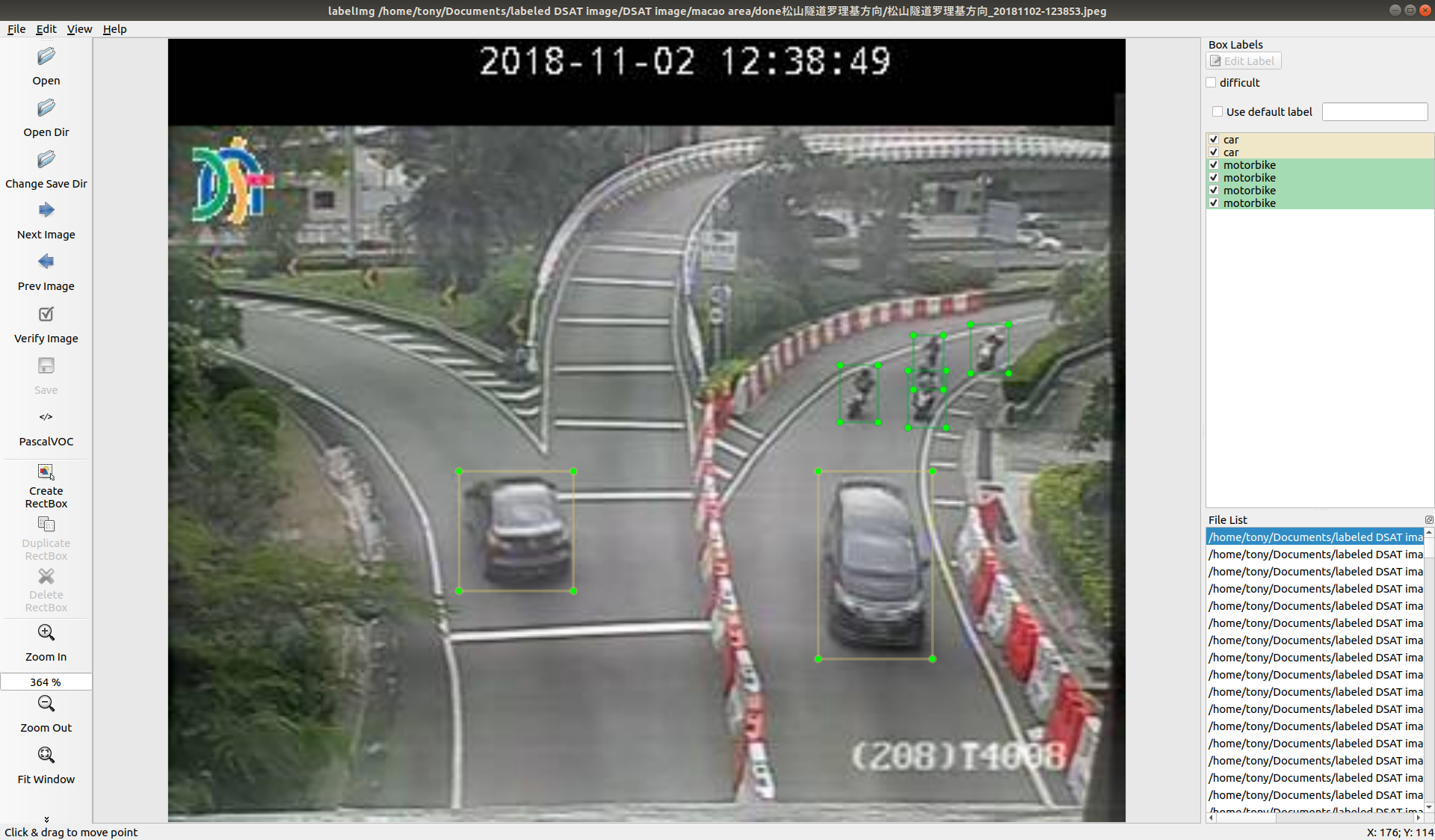
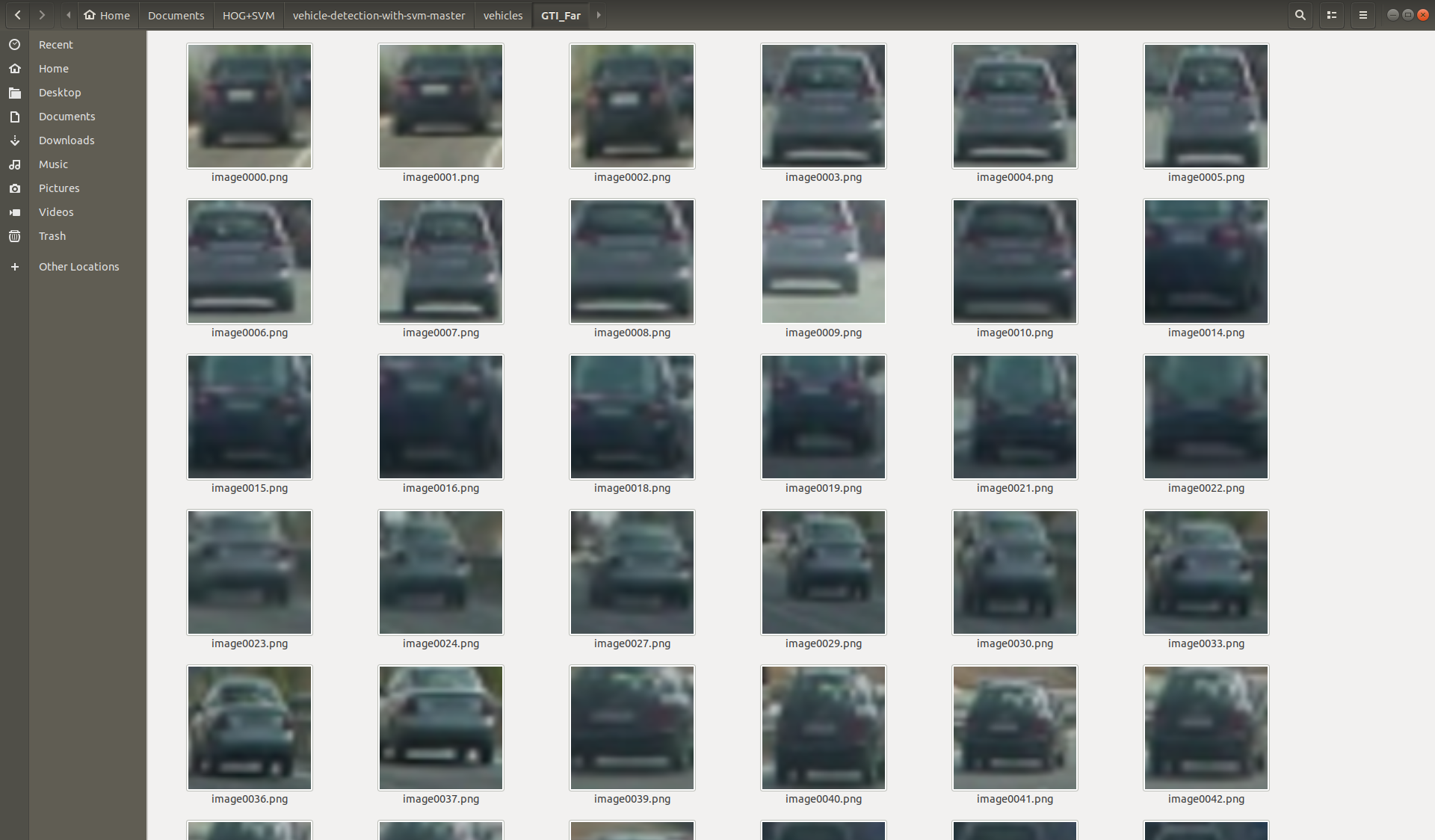


Figure 6 One example of labelled image

## Histogram of Oriented Gradients with Support Vector Machine

The first object detection algorithm implemented is HOG+SVM as a benchmark. This algorithm could be divided into two parts. The first part is to compute the histogram of oriented gradients of training images, another part is using support vector machine to realize linear classification.

The very first thing before using HOG and SVM was that finding datasets of cars (Figure 7) and non-cars (Figure 8). The two datasets are from GTI dataset [11] and KITTI dataset [12] and they both have size of 64 pixels by 64 pixels. The working principle of HOG and SVM is quite straightforward. First step is to define the size of cell which will scan the images. After that, the gradients will be computed inside each cell and then create histograms to record them. Also, the normalization will also be performed to get rid of variance of illumination changes. The next step would be computing HOGs for positive and negative training images and training linear SVM to learn how to classify them (Figure 9). The flowchart can be seen in figure 10.



**Figure 7 Car dataset**

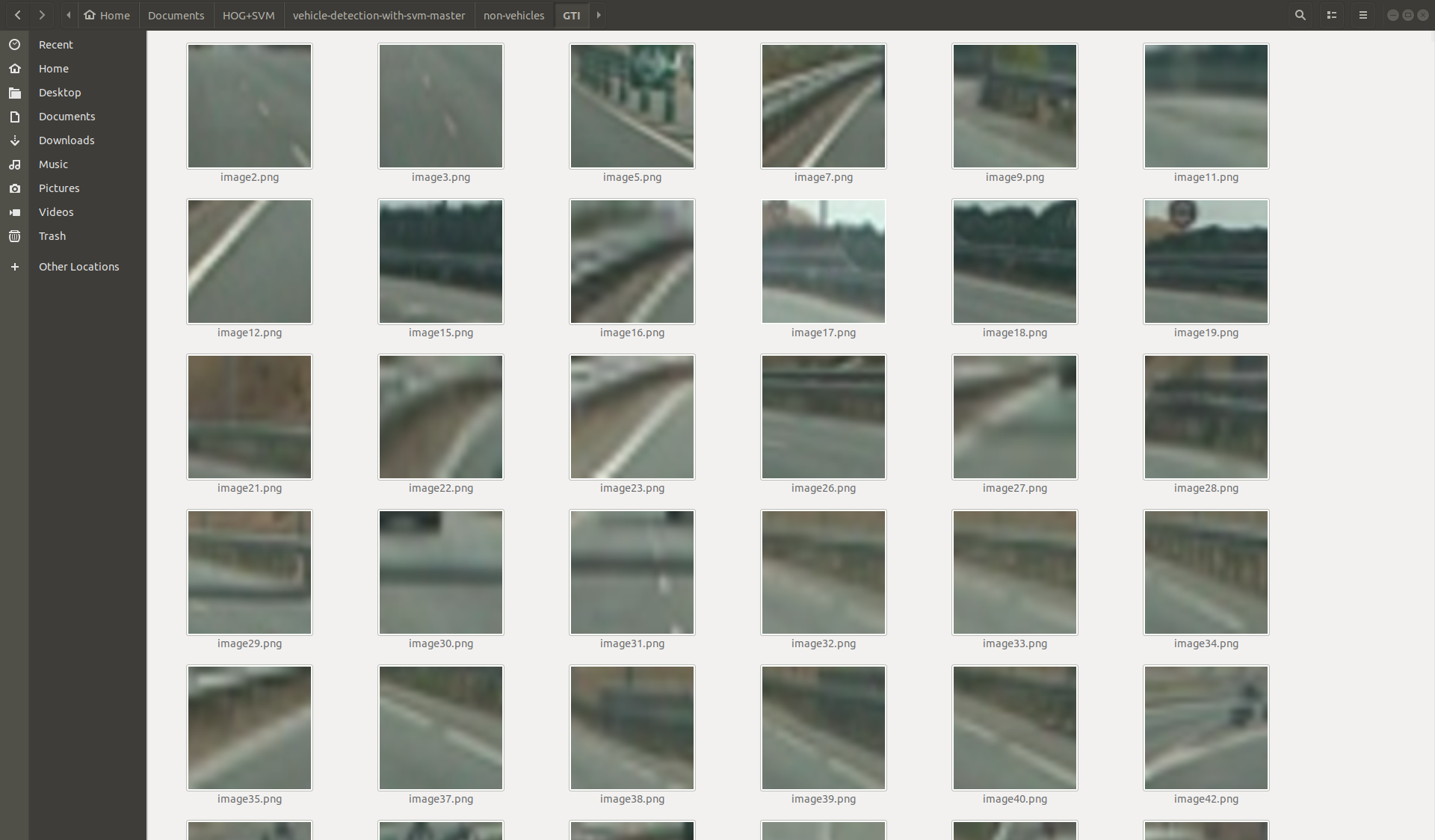


Figure 8 Non-car dataset



Figure 9 Linear SVM [18]

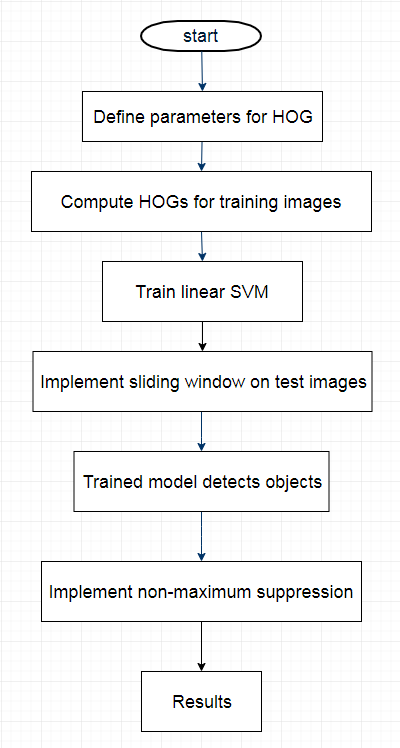


Figure 10 Flowchart of HOG+SVM

After that, the histogram of oriented gradients should be computed. Firstly, the cell was defined as size of 8 pixels by 8 pixels. Secondly, 64 (8x8) gradient vectors were computed and contributed to 9 orientations which represent 0-180 degree. Finally, in order to normalize gradients to ensure invariance of illumination change, 2 cells by 2 cells would be considered as a block to do normalization. Finally, the HOG is done (Figure 11.1 and Figure 11.2).

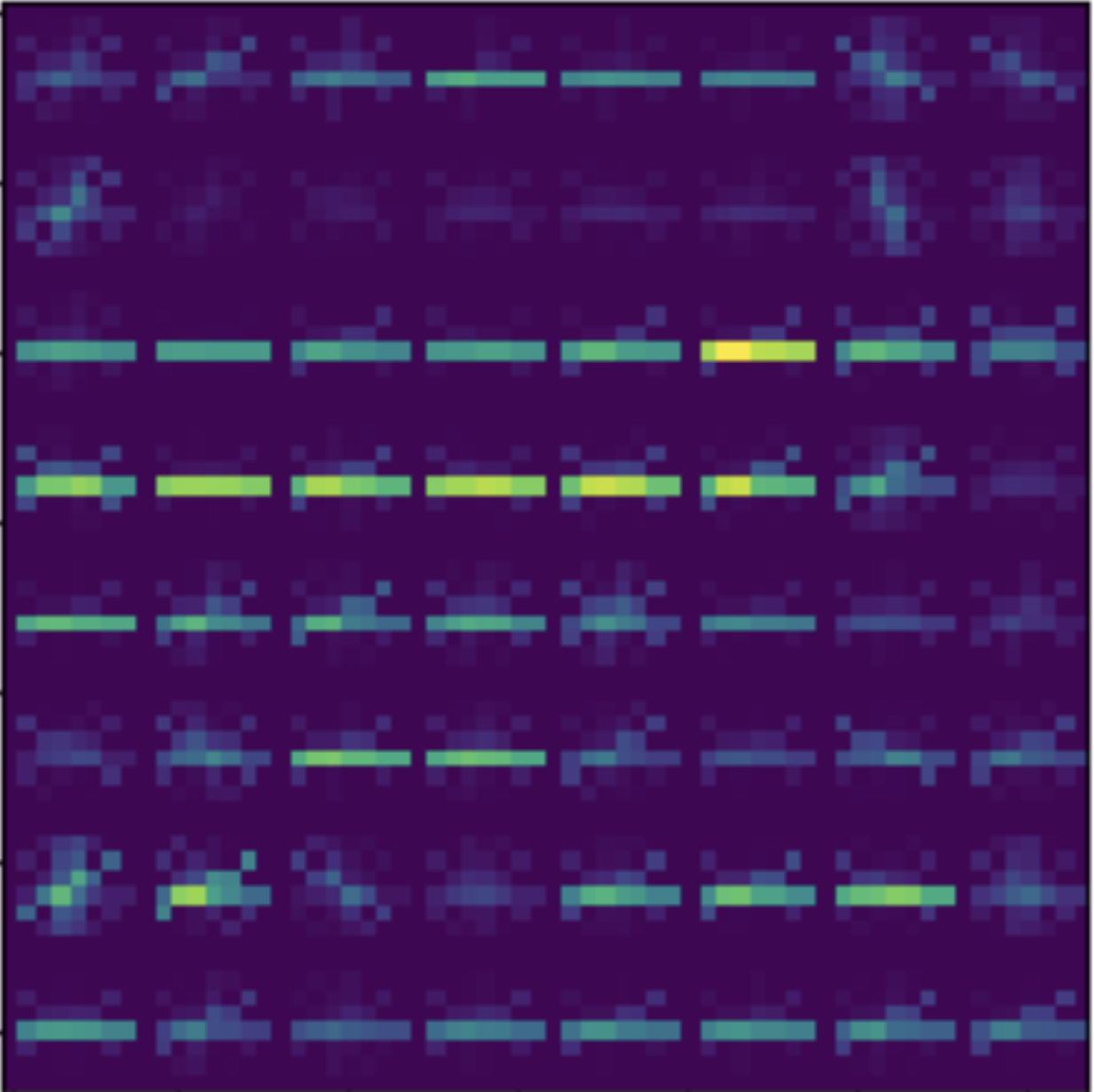


Figure 11.2 HOG of the image

Figure 11.1 Original image

Also, the spatial features of training images were also extracted by resizing them (64x63x3) into 32 pixels by 32 pixels (32x32x3) and then flattened into 1d array with size of 3072 (32x32x3). The main purpose of this is to give different colours in images.

Before feeding these two features into SVM, the last feature was going to extract is colour distribution within images, because the cars may have same colour but different shapes. In order to do so, the histogram of different channels would be extracted and computed.

The first part was done. The linear SVM should be trained by extracting these three features. After training the linear SVM with training dataset, 99.29% can be achieved as the prediction accuracy of the SVM to decide whether an image contains cars or not. The DSAT images would be tested whether it can detect cars or not. Before doing so, the sliding window should be defined for implementing sliding window search (Figure 12).

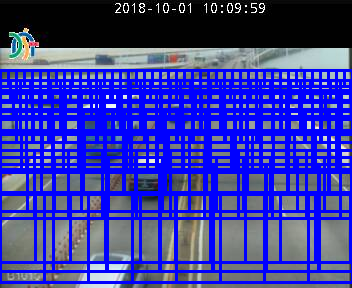


Figure 12 Sliding window

After that, The DSAT images are tested to detect cars on the roads. The results are shown below (Figure 13.1 and Figure 13.2). It is inevitable that there are bounding boxes problem which means that multiple boxes would be drawn surrounding the image. In order to solve that, the non-maximum suppression (NMS) was implemented. In NMS, all bounding boxes will be sorted firstly. Then the first bounding box will be picked. By computing the Intersection over Union (IOU), those bounding boxes with significant overlap ratio which exceed threshold will be ignored. The results are shown below (Figure 14.1 and Figure 14.2).

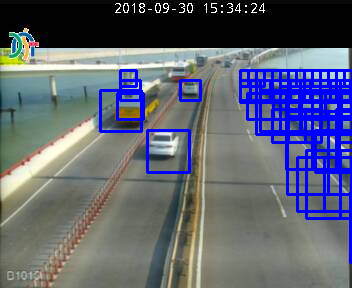
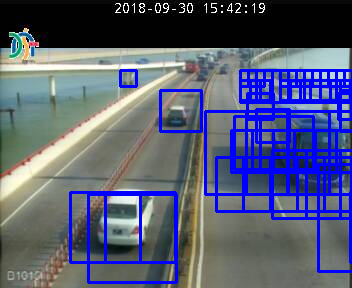
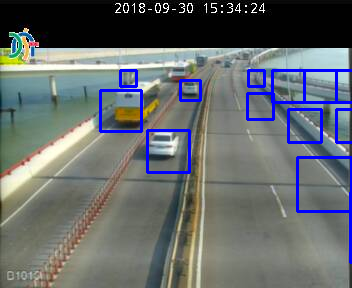
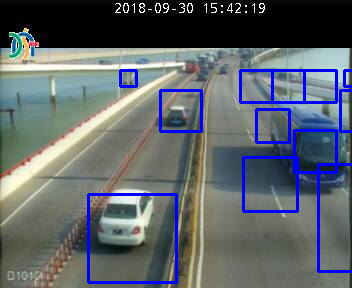


Figure 14.2 Test Image 2 using HOG+SVM with NMS

Figure 14.1 Test Image 1 using HOG+SVM with NMS

Figure 13.2 Test Image 2 using HOG+SVM

Figure 13.1 Test Image 1 using HOG+SVM

The non-maximum suppression really can solve the bounding boxes problem while there are also some false positive boxes need to be solved.

## YOLO

The next object detection algorithm implemented is YOLO which is the state-of-art real-time object detection system.

Because training a YOLO model is very time consuming and requires large volume of dataset, the pre-trained model was trained on COCO dataset [13] which can detect 80 classes.

Firstly, the YOLO v2’s architecture needed to be understood. The input is the number of ‘m’ images with the shape of 608 pixels by 608 pixels by 3 channels (m, 608, 608, 3). The output would be a list of bounding boxes with recognizing classes (confidence of class, the x coordinate of bounding box, the y coordinate of bounding box, the height of bounding box, the width of bounding box, 80 classes).

Secondly, the images will be divided into 19x19 cells and there are 5 anchor boxes are introduced with different sizes inside each cell. The anchor boxes would increase the accuracy of prediction. Therefore, the output shape would be (m, 19, 19, 5, 85).

Thirdly, because there are 80 classes the model will predict, we set a threshold to filter out those predictions with low confidence.

Fourthly, the non-maximum suppression would also be implemented to solve duplicated bounding boxes problem.

Finally, the images from DSAT would resize into 608x608x3. After that, the pre-trained YOLO model would be used to detect objects. The results are shown below (Figure 15).

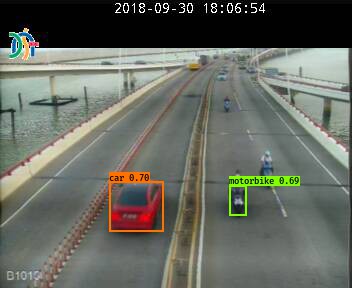


Figure 15 Test Image 1 using YOLO

However, sometimes, the YOLO cannot detect any cars on the roads (Figure 16).



Figure 16 Test Image 2 using YOLO

One of reasons is that the training images with the size of 608 pixels by 608 pixels, however, the resolution of images from DSAT is only 288 pixels by 352 pixels. The model was trained on clear images, but the test images are blurred which are even less than half of the training images. The two datasets come from different distributions. It would be undeniable that the results would be not so satisfied. Therefore, I also tested an image of 600 pixels by 800 pixels with similar camera angle but much larger resolutions. The result is much better (Figure 17).

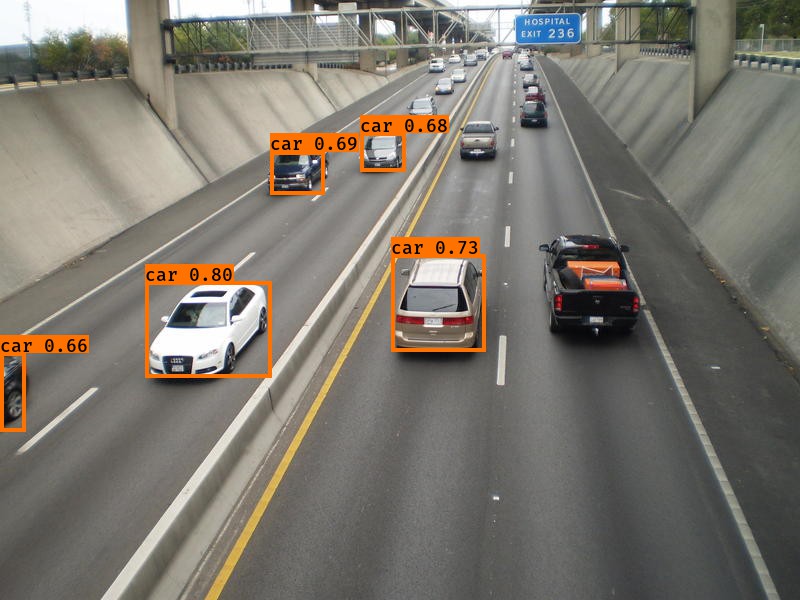


Figure 17 YOLO in clearer image

Another reason is that YOLO v2 is often struggling with small objects [14] and the cars on the images are quite small, the YOLO v3 should be tried out.

# On-going and Future Work

In this chapter, the on-going and future work is represented as followed. Also, the Gantt chart will be given (Figure 18.1 and Figure 18.2).

## HOG+SVM

Currently, the implementation of HOG+SVM is satisfying regardless of false positives bounding boxes in the test images. Next step will be how to improve the accuracy of prediction by reducing the bounding boxes.

One positive way would be using heat map to reduce the false positives. Another way will be adjusting the size of sliding window to fit the images from DSAT. These two methods should increase the accuracy of prediction.

## YOLO

In terms of YOLO, as discussing above, the accuracy may be affected by the huge different distributions of training data and test data. The course from Coursera [15] provides a solution by retraining the model with a dataset of blurred images. Therefore, the self-made labelled DSAT dataset with a large amount of images will be prepared for training purpose. Also, YOLO v2 is struggling with small objects and YOLO v3 would solve this problem.

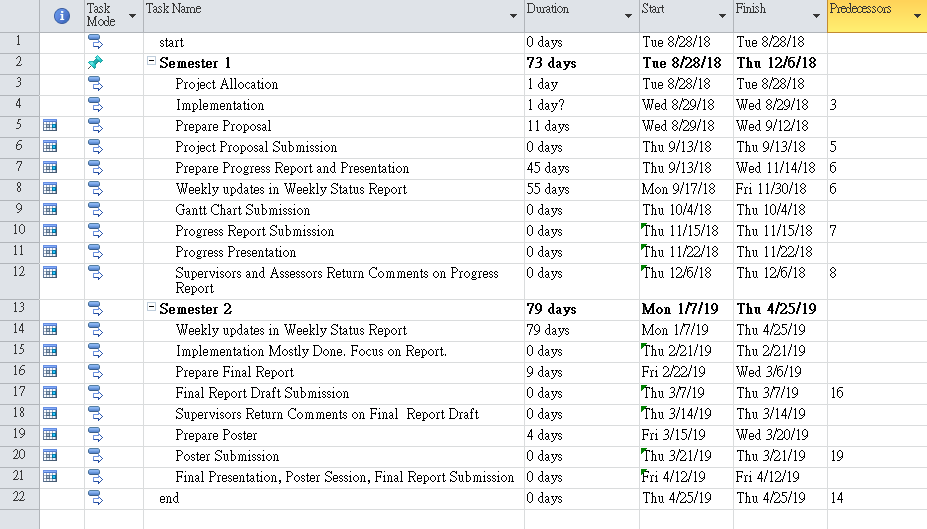


Figure 18.1 Gantt chart 1

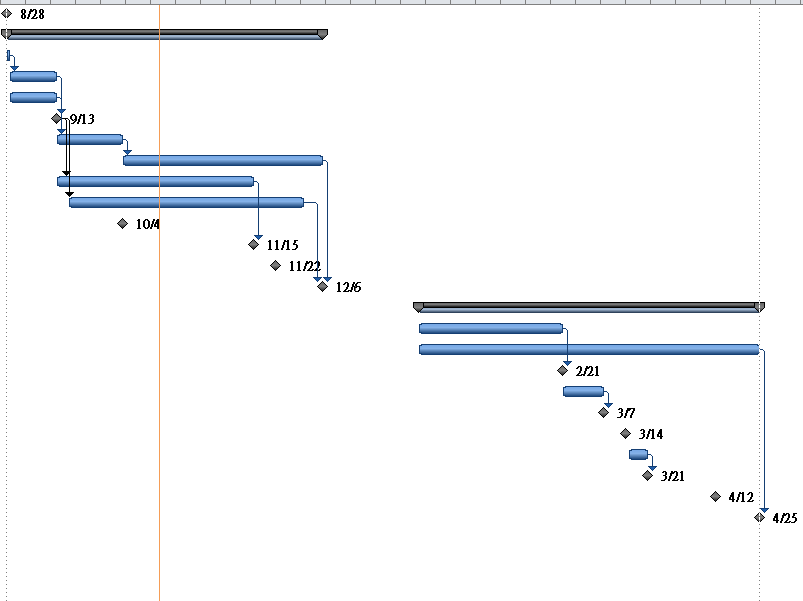


Figure 18.2 Gantt chart 2

# Conclusion

In this project, I have done literatures review about using images from traffic cameras to detect vehicles on the roads. By detecting and counting the number of cars, it can give clearer ideas of congestion level on certain roads.

So far, the histograms of oriented gradients with linear support vector machine have been implemented as a benchmark. The non-maximum suppression (NMS) also has implemented to reduce duplicated bounding boxes. The results are quite decent in spite of there are false positives in the prediction. Next step will be reducing the false positives as much as possible.

Another algorithm I have implemented is YOLO model, one of the state-of-art object detection models. The performance of this model should be quite decent compared to histograms of oriented gradients with linear support vector machine. However, currently, the prediction results are not satisfied to me. Sometimes, the model doesn’t even detect any cars in a given image. One of possible reasons is that the pre-trained YOLO v2 model is trained on the COCO dataset which are much clearer and have larger resolutions compared to testing images from DSAT. The distributions of these two datasets are too different to detect objects for models. Retraining the YOLO model with relative blurred images and using latest YOLO model should increase the accuracy of object detection.

During doing this project, I have met more than one challenges and I have to tackle them by myself. Also, because I have not done such projects before, I have to read papers and learn different algorithms by myself. I hope I will do better in the next semester.

References

[1] Transport Bureau of the Macao S.A.R. <http://www.dsat.gov.mo> [Nov. 11, 2018]

[2] TensorFlow. <http://www.tensorflow.org>. [Nov. 11, 2018]

[3] Scikit-learn. <https://scikit-learn.org/stable/index.html>. [Nov. 11, 2018]

[4] Keras. <https://deras.io>. [Nov. 11, 2018]

[5] Dalal, N., Triggs, B. Histograms of oriented gradients for human detection. In: Proc. CVPR 2005, vol. 1, pp. 886-893. <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1%467360>, 2005.

[6] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2147-2154, 2014.

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

[8] A. Atvar , Y. Artan ; Ş. Öztürk. Traffic density analysis using traffic camera images. 2017 25th Signal Processing and Communications Applications Conference (SIU) (pp. 1-4). Antalya, Turkey: IEEE., 2017

[9] Google Dataset Search. <https://toolbox.google.com/datasetsearch>. [Nov. 11, 2018]

[10] LabelImg. <https://github.com/tzutalin/labelImg>. [Nov. 11, 2018]

[11] Vehicle Image Database. <http://www.gti.ssr.upm.es/data/Vehicle_database.html>. [Nov. 11, 2018]

[12] A. Geiger, P. Lenz, C. Stiller and R. urtasun. Vision meets Robotics: The KITTI Dataset. International Journal of Robotics Research (IJRR), 2013.

[13] COCO. [http://cocodataset.org/#home](http://cocodataset.org/" \l "home). [Nov. 11, 2018]

[14] What’s new in YOLO v3? <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>. [Nov. 11, 2018]

[15] Improving Deep Neural Networks: Hyperparameter tuning, Regularization and Optimization. <https://www.coursera.org/learn/deep-neural-network>. [Nov. 11, 2018]

[16] J. Wan, Y. Yuan and Q. Wang. Traffic congestion analysis: a new perspective. IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 1398-1402). New Orleans, LA, USA: IEEE, 2017

[17] Traffic Cameras from the City of Calgary. <https://data.calgary.ca/Transportation-Transit/Traffic-Cameras/k7p9-kppz>. [Nov. 11, 2018]

[18] File:Linear-svm-scatterplot.svg. <https://commons.wikimedia.org/wiki/File:Linear-svm-scatterplot.svg>. [Nov. 11, 2018]