MULTIPLE ROAD USERS DETECTION AND TRACKING SYSTEM IN URBAN MIXED TRAFFIC SCENES

CHAN ZHEN YU

UNIVERSITI SAINS MALAYSIA

2020

MULTIPLE ROAD USERS DETECTION AND TRACKING SYSTEM IN URBAN MIXED TRAFFIC SCENES

by

CHAN ZHEN YU

Thesis submitted in fulfillment of the requirements
for the Degree of
Master of Science

November 2020

ACKNOWLEDGEMENT

First and foremost, I would like to deliver my sincere appreciation to my supervisor, Prof. Ts. Dr. Shahrel Azmin for his supports, advices, commitment and guidance throughout my master's degree. I would be a great challenge to finish the project within the allocated time frame without his dedication in steering this effort in the correct direction.

Besides, I would also like to express my thankfulness my fellow comrades, Bernard Cheah Jun Kai, who was also under Prof. Ts. Dr. Shahrel Azmin's supervision, for all the help and assistance I received throughout my research.

On my personal side, I would like to express gratitude to my parents for their unconditional support and comprehension given throughout my research. I could be difficult and tough journey without the mental and emotional given by them. Thank you for being there with me all the times. Without them, there would be hard to finish my thesis successfully.

It has not been an easy journey here to complete my master's degree. I would also like to express thankfulness to all my beloved friends for their supports. Their kind assistance and companionship have encouraged me to conquer every single challenge with perseverance and bravery.

Finally, a big shout out to everyone who had assisted me, either directly or indirectly, along my way in completing this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Θ	Parameter in neural network
y	Neural network input
$\hat{y}$	Neural network output
i	Label of training samples
x	Training data input
N	Number of training samples
m	Mini-batch size
k	Kernel size
u	Bounding coordinate in x-axis
v	Bounding coordinate in y-axis
h	Height of bounding box
t	Mahalanobis threshold
d	Mahalanobis distance
r	Appearance descriptor
T	Track
U	Unmatched detection
L	Label detection
g	Number of objects in ground truth
mme	Mis-match
Θ	Parameter in neural network
α	Learning rate
γ	Aspect ratio of *u* and *v*
λ	Hyperparameter of association

LIST OF ABBREVIATIONS

ANN	Artificial Neural Networks
CFT	Correlation Filter-based Tracking
CNN	Convolutional Neural Networks
DeepSORT	Deep Simple Online Realtime Tracking
DPM	Deformable Part-based Model
EKF	Extended Kalman Filter
GMM	Gaussian Mixture Model
HCI	Human-Computer Interaction
HSC	Histogram of Shearlet Coefficients
HOG	Histogram Oriented Gradient
IOU	Intersection Over Union
KCF	Kernelized Correlation Filter
KLT	Kanade-Kucas-Tomasi
LBP	Local Binary Pattern
mAP	Mean Average Precision
MCMC	Markov Chain Monte Carlo
MIO-TCD	MIOvision Traffic Camera Dataset
MLE	Maximum Likelihood Estimation
MOT	Multiple Object Tracking
MOTA	Multiple Object Tracking Accuracy
MOTP	Multiple Object Tracking Precision
MS	Mean-Shift
MSC-HOG	Multi-Size Cell HOG

PCA	Principal Component Analysis
R-CNN	Regional-based Convolutional Neural Networks
ReLU	Rectified Linear Unit
RPN	Regional-based Convolutional Neural Network
SIFT	Scale-Invariant Feature Transform
SGD	Stochastic Gradient Descent
SP-FRCN	Stereo-Proposal based Fast R-CNN
SS	Selective Search
SVM	Support Vector Machine
YOLO	You Only Look Once
YOLOv3	You Only Look Once Version 3

SISTEM PENGESANAN DAN PENJEJAKAN PELBAGAI PENGGUNA JALANRAYA DI DALAM SENARIO TRAFIK CAMPURAN BANDAR

ABSTRAK

Penggunaan teknologi analisis video untuk kawalan dan pemantauan lalu lintas semakin popular dalam tahun-tahun kebelakangan ini. Teknologi analisis video boleh membantu kawalan dan pengawasan trafik dengan menyarikan data trafik seperti pengiraan kenderaan dan pengelasan kenderaan daripada video trafik. Data-data tentang pengguna jalan raya yang dihasilkan daripada video trafik akan memberikan manfaat kepada perancang trafik. Hal ini membuktikan kesan analisis video dalam pengawasan trafik. Walaubagaimanapun, penjejakan pengguna jalan raya di bandar masih mencabar kerana penampilan pengguna jalan raya yang berbeza. Untuk mengatasi masalah seperti perubahan identiti dan pengelasan yang salah, kaedah yang dikenali sebagai 'City Tracker' yang menggabungkan Maximum Likelihood Estimation (MLE), YOLOv3 dan DeepSORT dicadangkan dalam penjejakan trafik di bandar. City Tracker menjangkakan koordinat sempadan yang berpotensi daripada keputusan YOLOv3 dan DeepSORT. Ia membanding dengan koordinat sebenar untuk mengatasi masalah pengelasan salah dan perubahan identiti. MLE mencadangkan pengelasan berasaskan londar untuk menyelesaikan pengelasan yang salah berasal daripada sudut penglihatan. Cara tersebut diuji dengan set data Urban Tracker. Penilaian prestasi pengesanan dan penjejakan menunjukkan bahawa City Tracker berjaya meningkatkan Ketepatan Pelacakan Pelbagai Objek (MOTA) daripada 0.3505 ke 0.3793 (8.28%) dan meningkatkan Pelbagai Objek Penjejakan (MOTP) daripada 0.6245 ke 0.6442 (3.15%). MLE meningkatkan Kadar Ingat daripada 0.7032 ke 0.7838 (11.46%) dan Ketepatan daripada 0.7214 ke 0.8334 (15.53%) dalam prestasi pengelasan berbanding dengan sistem yang tidak menggunakan MLE.

MULTIPLE ROAD USERS DETECTION AND TRACKING SYSTEM IN URBAN MIXED TRAFFIC SCENES

ABSTRACT

Video analytic technology in traffic control and monitoring is getting more attention in recent years. This is because video analytic technology can perform traffic surveillance to extract traffic information such as vehicle counting and classification from video sequence. Large amount of road user data can be generated from video and this data would benefit for traffic planner. This provides the impact of video analytic in traffic surveillance. However, multiple road users tracking in urban traffic remains challenging because of large variation of road user appearance. To overcome the problems of multiple object tracking in mixed urban traffic, which are mis-detection, frequent ID switches and mis-classification, a system known as City Tracker, which incorporates Maximum Likelihood Estimation (MLE), YOLOv3 and DeepSORT is proposed in mixed urban traffic. City Tracker predicts the potential bounding box coordinates from the result of YOLOv3 and DeepSORT, then matches with the latest actual bounding box to overcome the mis-detection and frequent identity switch. On the other hand, MLE provides trajectory-based classification to solve mis-classification. This solution is tested with Urban Tracker dataset based on detection and tracking performance. The performance evaluations show that implementation of City Tracker increases Multiple Object Tracking Accuracy (MOTA) from 0.3503 to 0.3793 (8.28%) and Multiple Object Tracking Precision (MOTP) from 0.6245 to 0.6442 (3.15%) which are calculated from Precision and Recall as the evaluation metrics. MLE improves Recall from 0.7032 to 0.7838 (11.46%) and Precision from 0.7214 to 0.8334 (15.53%) in classification performance, which is better than conventional YOLOv3 and DeepSORT that do not consider City Tracker.

CHAPTER 1

INTRODUCTION

1.1 Introduction

In recent years, a great deal of interest regarding computer vision and machine vision research area has attracted attention because of rapid development of high processing computers, popularity of widespread high resolution and low-cost vision sensors like CCD or CMOS, as well as the growing demand for automatic video analytic. The application of intelligent camera systems is getting popular in this decade. A vision-based automatic video tracking technology promises to be the important role for a large variety of domains such as automotive driving assistance system (Sochor, 2014), transportation traffic video analytics (Jun *et al.* 2008), automatic video surveillance and shopper behaviour video analytic.

Figure 1.1: Applications of Computer Vision in Traffic Surveillance Area

Automotive driving assistance system can assist people drive safely by detecting and tracking the vehicles and pedestrians on streets. By giving assessment of various of dangers through vision-based analytic (Sochor, 2014), the system can give the driver proper instructions to avoid accident, thus keeping people shielded in the presence of autonomous cars. Transportation traffic video analytics play an important role in getting in-depth insights into the daily traffic, and these can help in curbing congestions by routing the information to the traffic management console. As more vehicles hit the roads every day, traffic congestion becomes a major issue in many cities around the world. It becomes important to have fast and intelligent traffic management solution that can help top curb traffic congestion.

Traffic control and monitoring (Jun *et al.*, 2008) using video analytic has drawn increasing attention due to the significant advances in the field of computer vision. In general, systems developed for traffic surveillance purposes aim at having an understanding of real time traffic conditions (Coifman *et al.*, 1998), which require the analysis of the static environment and the detection of static or moving obstacles. Image processing and computer vision techniques (Li *et al.*, 2009) are being applied to the analysis of video sequences of traffic in order to extract traffic information such as vehicle counting, vehicle classification, speed measure, traffic flow and etc, as shown in Figure 1.2.

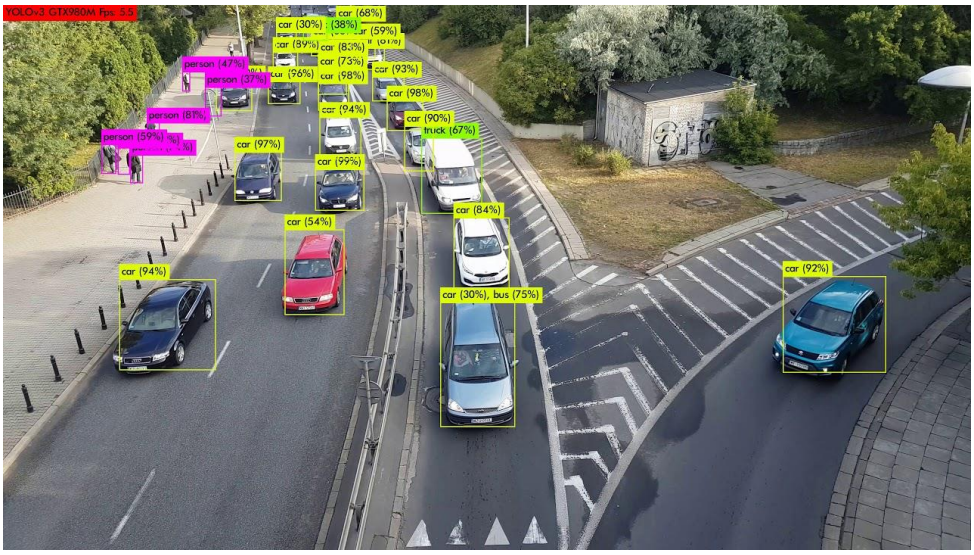

Figure 1.2: Traffic Video Analytic System (Redmon and Farhadi, 2017).

Video analytic technology is also fostering new research in transportation engineering such as the study of behaviour and the safety of cyclists, cars, pedestrians and other road users. Video analytic technology can perform road safety diagnosis (Huang *et al.*, 2012) from the observation of road users interactions instead of waiting occurrence of accidents. This technology would allow road user to acquire the road condition information at lower cost, easier and more accurate than manual data analyst. Besides, large amount of road user data can be collected using video analytic technology, which would enable to further analyse the road user behaviour by learning the vehicle trajectories, vehicle speed and vehicle model (Jodoin *et al.*, 2016).

As more and more surveillance cameras are deployed in highway (Tamersoy and Aggarwal, 2010) and urban traffic, traffic video analytic system has come to a stage where it plays an important role. Human operator could miss a potential abnormal event as the amount of information and data that have to be handled is high. Traffic video analytic supports operator by automatically detecting abnormal event in traffic. The main concept is to aid human operators in observing video data. This can allow online and post-event detection of events of interest, which is useful for traffic

management. Therefore, development of a traffic video analytic system is highly significant to solve the data analytic issue in transportation and traffic efficiently. The system that can constantly follow targets over the entire tracking process is very helpful in transportation. With improved robustness against various challenges such as mis-detection and mis-classification, a traffic video analytic system can exploit advantage from vehicle behaviour analysis, traffic safety and science investigation.

1.2 Problem statement

There are two important categories of applications of vehicles tracking system which are highway traffic scene and urban traffic scene (Saunier and Sayed, 2006). Tracking vehicles in highways, as shown in Figure 1.3 (Coifman *et al.*, 1998), is simpler than in urban areas because there are fewer type of road users which contain only various sizes motorized vehicles and no pedestrians, limited change of vehicles orientations and few known exits and entries points. Tracking vehicles in mixed urban traffic scenes, as shown in Figure 1.4, is usually more challenging because of occurrence of occlusions. The occlusion risk between the vehicles rises when the traffic is slow because the inter-space decreases.

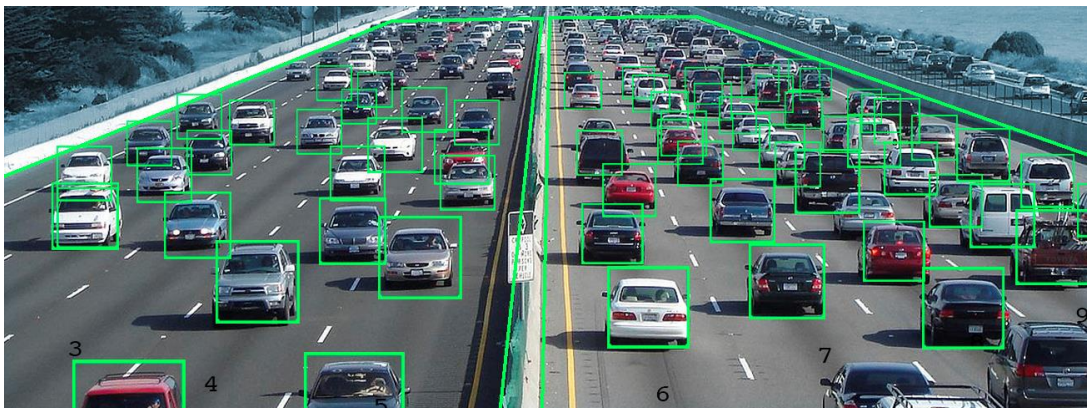

Figure 1.3: Highway Traffic Scenes (Coifman *et al.*, 1998).

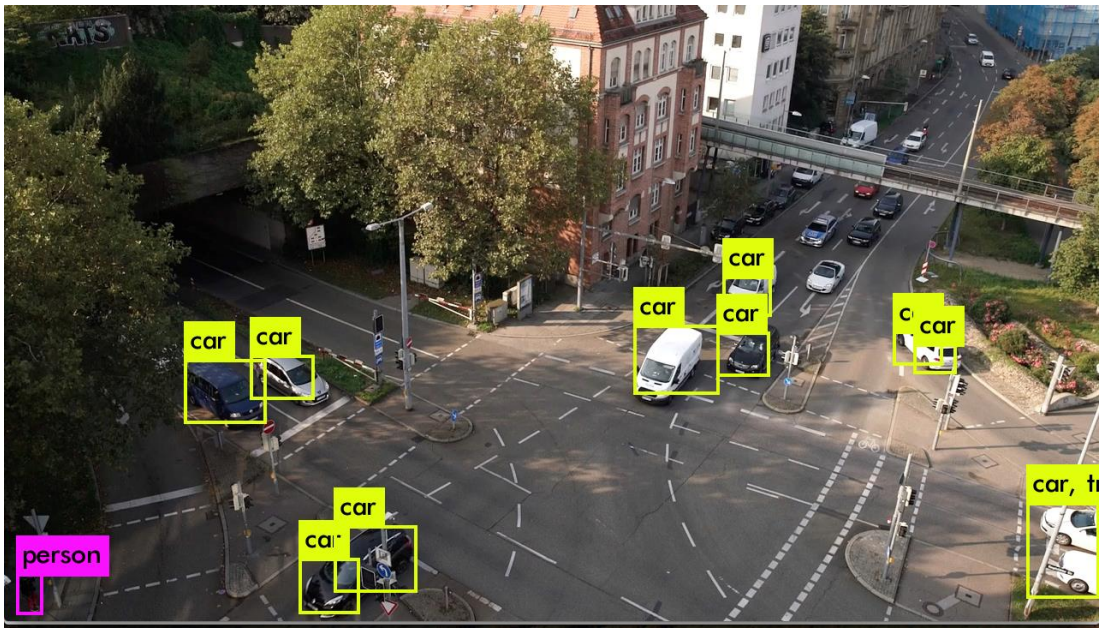

Figure 1.4: Urban Traffic Scenes (Jodoin *et al.*, 2014).

Moreover, urban traffic scenes sometimes include cyclist and pedestrians, and more complex tracking trajectories including turning intersections of vehicles, parking and stopping and many other entries and exit point. Variation of urban tracking (Jodoin *et al.*, 2016) is very challenging and cannot be easily addressed by reason of the large road users variety for instance pedestrians, cars, cyclists, truck, vans and others. All these objects are different in shape, appearance and colour. Pedestrians are similar in shape and their appearance, but they vary differently in particular the dressing colour and they are non-rigid. Vehicles contain different colours and shapes, in particular when their pose changes as they move throughout the video scene and they turn at intersections. As a result, there are some solutions that only exploit motion information to discover the road users, either using background subtraction or optical flow. However, this solution cannot classify vehicle type correctly and there are mis-tracking issue when two objects are overlapping.

The limitation of traditional background subtraction is occurrence of occlusion during tracking (Ooi *et al.*, 2018), which increases the risk of mis-tracking. Tracking-

by-detection can provide real time detection, classification and tracking. Nevertheless, the performance of tracking and classification is highly relied on the detection result. In mixed urban traffic, there is a lot of uncertainty and inconsistent of object appearance and motion due to road user turning intersection and occurrence of occlusions. This is challenging for object detection. The overall performance for classification and tracking are significantly affected once the object disappeared during tracking. A solution that could improve consistency of object detection can help reduce problem in mixed urban traffic.

Besides, frequent switching of classification result from detector always happen in mixed urban traffic. This is because of the change of road user orientation, camera viewing angle and object turning intersection. The consistency of object classification result should be improved to reduce the false positive issue that happens in mixed urban traffic. To build a system for long-term tracking and detection of an aprior unknown number of road users with random movement in an overlapping outdoor condition, a large number of detectors for different types of road users and their primary poses are required to track all possible road users. A tracking method which tracks the road users based on the appearance and motion, then recovers identities after occlusions would be advantageous in urban traffic scenes (Wojke, Bewley and Paulus, 2018). Besides, higher consistency of classification result from the detector is important to ensure the performance of vehicle classification and to reduce the false positive of classification result issue while the object is under tracking.

1.3 Objectives

The present study contains three important objectives, which are listed below

i. To improve the multiple objects tracking performance in mixed urban traffic scenes.
ii. To compare and validate the multiple tracking performance between tracking-by-detection and motion information.
iii. To provide the solution to reduce false positive of classification result during tracking.

1.4 Contribution of the study

The main contribution of this research is a multiple road user tracking solution to track an unknown number of road users within crowded, complex, mixed urban traffic scenarios on the fundamental of tracking-by-detection solution.

i. A solution is implemented to the system for multiple object tracking-by-detection with the benefit of being resistant to divergence and superior robustness. Instead of typical tracking-by-detection solution such as combination of YOLOv3 and DeepSORT, City Tracker proposes prediction of bounding box position when the road users are not detected to increase the overall tracking performance in mixed urban traffic scenes.
ii. Maximum Likelihood Estimation (MLE) is implemented to estimate the highest potential classification result to reduce the frequent switching of classification result and improve the overall classification result accuracy.
iii. A comparison between traditional background subtraction tracking, tracking-by-detection solution and City Tracker based on tracking-by-detection to determine the optimum solution in mixed urban traffic based on multiple tracking performance.

1.5 Scope of study

The performance of Multiple Object Tracking in mixed urban traffic scenarios is evaluated based on Multiple Object Tracking Accuracy (MOTA) and Multiple Object Tracking Precision (MOTP). All threshold values, initial conditions, boundary condition and parameters are all defined by using Python code environment. This code is developed under KERAS deep learning framework, which is an open-source neural-network library written in Python.

The overall tracking performance of background subtraction and tracking-by-detection are investigated in this research. Tracking-by-detection method has two main components which are object detection and object tracking. The performance of different tracking-by-detection method which are YOLOv3+DeepSORT, YOLOv3+DeepSORT+City Tracker, combination of RFCNN with tracker and Baseline tracker are investigated under Urban Tracker Dataset (Jodoin *el at.*, 2014).

The pretrained weightage of YOLOv3 (Redmon and Farhadi, 2018) is fined-tuned by using MIO-TCD dataset (Luo *et al.*, 2018) using transfer learning. The purpose is to provide label detection to 11 classes: articulated truck, pedestrian, bus, work van, bicycle, motorized vehicle, non-motorized vehicle, motorcycle, pickup truck, single unit truck and car. Different layers detection strategy is useful to solve the problem of detecting small objects. The up-sampled layers which is concatenated with the previous layers assist preserve the fine-grained features and help in detecting tiny objects.

1.6 Thesis outline

This thesis is divided into five main chapters. Chapter 1 is dedicated to providing an overview of the background and research details. A literature review on past and on-going research on related topics is given in Chapter 2. Chapter 3 provides

the proposed architecture of the system and experimental set-up details. The results obtained from experiments are discussed and displayed in Chapter 4. Finally, Chapter 5 provides the overall research summary and the conclusions. Recommendations for further studies are suggested as well in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

In recent years, numerous works related to multiple-object tracking, detecting moving and motionless objects have been proposed. Different approaches depend on different sensors have been employed to detect objects such as monocular and stereo camera, lidar and radar. Among these options, vision sensors are preferred because of their capability to obtain a high resolution perspective view from the scene with colour and texture information (Gerónimo *et al.*, 2010; Sochor, 2014; Rawat and Wang, 2017). This is because vision techniques are more cost-effective than other methods and it can process many other works for instance lane detection road user classification and detection of traffic sign.

Object tracking is one of the important tasks among the computer vision field. The goal is to determine the precise position of one or more target objects in complex and dynamic scenes. There are a lot of common applications such as surveillance, self-driving robotics, human-computer interaction (HCI) and industrial automation that require object tracking from computer vision as its fundamental technology. Recently, there have been great improvement on classification, detection and tracking in application field but this task is still difficult due to several factors such as illumination, occlusion, scale variation, shape deformation and fast motion.

2.2 Motion based approach

Object detection in computer vision field can be divided into two main categories which are motion-based approach and appearance-based approach. In year 2004, Hu *et al.*(2004) categorized motion detection methods into three sub-classes which are frame differencing, background subtraction and Gaussians mixture model

(GMM) (Zivkovic, 2004), which is also known as adaptive background subtraction. Frame differencing is a pixel-wise differencing method between two or three consecutive frames in an image sequence to detect regions corresponding to moving object while the concept in background subtraction is to subtract the current image from reference background image. This method works well in stationary cameras without any illumination changes. GMM detects the moving regions in a group of pixels whose distribution does not fit Gaussian distribution of background pixels. This method can handle illumination changes, slow and repetitive motion effectively.

Motion based detection can be applied in real time applications because they are fast and require less processing power. Li *et al.* (2009) proposed real time pedestrian detection based on GMM (Zivkovic, 2004). There are some use cases to combine motion based and appearance based approaches to speed up and augment the detection such as Fujimoto and Hayashi (2013) and Dahiya *et al.* (2016).

2.3 Appearance based approach

Traditional computer vision performance replies on the power of selected features applied in detection method. Basically, there are some feature types which have been mostly applied to detect road users which are simple template features, part-based features and geometric features. All these feature types use appearance properties of target, for instance, shape, intensity and texture.

2.3.1 Traditional Appearance based approach

Cho *et al.* (2010) proposed a road user detection solution, which is based on Deformable Part-based Model (DPM) (Felzenszwalb *et al.*, 2010) by applying Principal Component Analysis (PCA) of Histogram of Oriented Gradient (HOG) Linear Support Vector Machine (SVM) (Cortes *et al.*, 1995), and Extended Kalman Filter (EKF). Jung *et al.* (2012) proposed Multi-Size Cell HOG (MSC-HOG) which is

based on improved HOG to detect road user. Dahiya *et al.* (2016) proposed a method that used both motion based, and appearance based to detect road user. An improved adaptive Gaussian Mixture Model (GMM) (Zivkovic, 2004) was applied to differentiate between moving and static object in video frames. Three features which are HOG, Local Binary Pattern (LBP) (Wang *et al.*, 1990) and Scale-Invariant Feature Transform (SIFT) (Lowe, 2004) were separately used to classify the moving object returned by background subtraction whether they are road user and their performance were compared. In 2017, Lin *et al.* (2017) proposed a side-view bicycle detection method based on the geometric relationship of two wheels and two triangles in the side-view of bicycle images. In prospective projection, the object is triangle and ellipse, therefore, Canny edge detector is applied to extract the edge information. The information is then classified by using Hough Transform to detect triangle and ellipses. Based on the geometric relationship between triangles and ellipses, bicycle frames and two wheels are found, and the geometric model validation is applied to connect all the parts of bicycle.

However, this method is not appropriate in mixed urban traffic video because of lack of rotation variant and lack of robustness under noisy and small images. Fujimoto *et al.* (2013) proposed a bicycle detection method by combining motion based and appearance-based method. In this method, an optical flow algorithm is applied to detect the moving region in video frames and ellipses approximation is applied to estimate the bicycle tire in moving regions. By evaluating the width and angle of bicycle tire, this method can estimate the bicycle traffic direction but it is not appropriate in occurrence of occlusions. Moreover, this method is only limited for bicycle detection instead of road user detection in mixed urban traffic.

2.3.2 Convolutional Neural Network approach

Recently, there is great improvement of appearance-based approach detection due to implementation of deep learning. Traditionally, system takes a classifier for that object and evaluate it at various scale and locations in a test image. By implementing deep learning, manual feature extraction is removed because deep learning network structure covers both feature extraction and classification. The overall performance of detection and classification are improved significantly compared to traditional feature extraction and narrow network. Moreover, the increased computing power and substantial amounts of training data in this information era have led step forward of deep learning architecture in object classification and object detection. The analogy of deep learning is that the rocket engine is the deep learning models and the fuel is the large amounts of data that we can feed to these algorithms.

In deep learning, a Convolutional Neural Network (CNN) is a class of deep neural networks, which is commonly applied to analysing visual imagery. CNN has been widely applied among object detection and object classification algorithm of traffic surveillance. CNN method outperformed a lot of traditional methods such as HOG (Dalal and Triggs, 2005) and LBP (Wang and He, 1990) which have been the state-of-the-art for many years. However, CNN uses deep learning architecture which is powered by massive amounts of data. Besides, it requires high computing power, in contrary to traditional machine learning.

Computer vision that applies deep learning architecture in form of CNN is usually a sequence of convolutional, activation and pooling layers to the actual fully-connected network. The benefit of CNNs are weights of the convolutions learned automatically to minimize a specific loss function. This method eliminates the procedure of manual feature selection and leads to convolutional image features.

Traditional manual feature selection such as HOG, MLBP and Histogram of Shearlet Coefficients (HSC) encode very low-level characteristic of the object. Thus, they are not able to classify well among the different labels. However, CNN methods can construct a representation in hierarchical manner with increasing order of abstraction from lower to high level of neural networks.

CNNs are feedforward neural networks which can be explained as models that learn visual filters to recognize higher level image features. The idea inspiration of this architecture comes from the mechanism of biological visual perception. Feedforward neural network represents a set of modelling tools that have been proven to be very successful in pattern recognition, detection, classification, regression and other tasks. The main property of a feedforward network is that information flows through the network along a single direction and the network does not contain any feedback. The neurons are arranged in layers so that the network has a dedicated input layer, output layer and potentially some hidden layers, providing a systematic method of calculating the activations of the output layer, given the input layer and the weights and biases of all intermediate layers. Figure 2.1 shows an example of feedforward fully connected layers with N hidden layers.

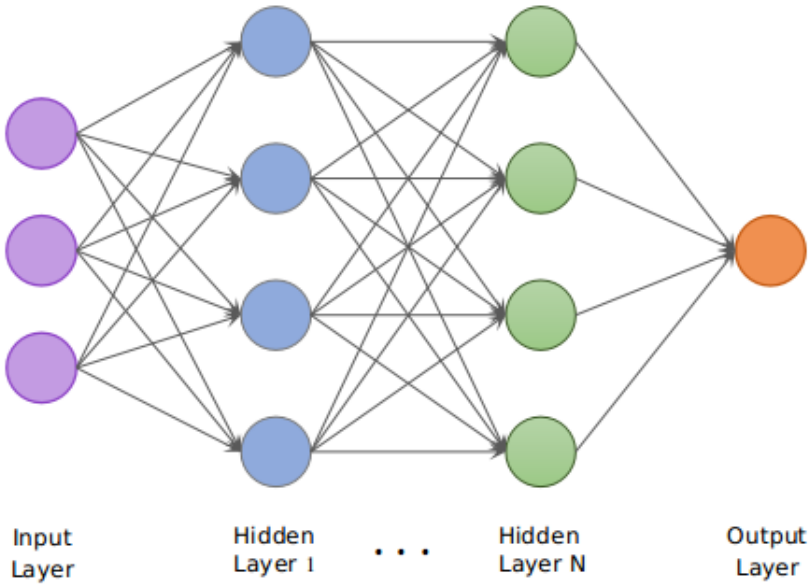

Figure 2.1: Example of fully connected layer feed forward neural network with *N* hidden layers.

CNN networks are composed of neurons with learnable weight and biases, just like Artificial Neural Network (ANN). Each neuron receives some input, performs a dot product and optionally follows it with an activation function. The CNN architecture is typically composed of several hidden layers which provides them a characterisation of deep layers and, therefore, the research work on CNNs fall under the domain of deep learning. The CNN network computes a mapping function that relates image pixels to a final desired output. For instance, the input is assumed to be an RGB image which consists of three channels, corresponding to red, green and blue colour intensity values. Consecutive layers of CNN may consist of even more channels known as feature maps. The number of feature maps increase through the layers of a CNN while the spatial dimension of them decreases until reaching the desired output.

Back propagation is a technique to propagate errors in the neural network back through the feedforward architecture and adapt the weights. There are two steps to train a neural network with backpropagation which are feedforward and

backpropagation step. A training case is classified in the feedforward step by using the current neural network while a classification error is computed and propagated back using the neural network in the backpropagation step. Therefore, in the training process, these require having predetermined desired outputs for given input data which can be compared to the actual output ANN. The desired output *y* along with the actual output $\hat{y}$ is passed to a differentiable cost function which is minimized by adjusting the parameters which are bias and weightage of the network.

The procedure of passing data through the network, calculating the cost and adjusting the parameters continues until the network has reached an acceptable accuracy when evaluated on the validation data set which is separated from the training data set. By applying gradient descent, backpropagation propagates the gradients of the cost function with respect to the parameters back through the network using the chain rule. The weights are updated based on the learning rate, error and gradient of the activation. Generally, the training of the network in CNNs is usually done using an optimization algorithm called stochastic gradient descent (SGD). While only one sample of input data, desired output and actual output are required to calculate the gradients for all parameters of the network, it is common practice to include several samples called mini-batch and take average of the obtained gradients. SGD is a process of randomly selecting samples from the training set, computing the gradients and then updating the parameters of the network.

The amount of training done is normally measured in epochs, which is defined as the number of times all training samples have been used to update the network parameters. If the number of available training samples is *N*, one epoch is completed after using $\frac{N}{m}$ mini batches for training.

Activation functions are applied to execute non-linear mappings of the input data and are typically used as the element-wise to all neurons in a hidden layer. In deep learning network architecture, there are some types of activation functions which are Sigmoid, Rectified Linear Unit (ReLU) and Softmax. Among these activation functions, ReLU function is commonly applied as an activation function for intermediate layers of neural network while Softmax layer is generally applied in last layer. The resulting feature vectors are corresponding to the probability distribution of classes by applying a softmax function as an activation function.

Convolutional layers have multiple filters that are determined based on their weights. The number of filters and the size of kernel, amount of padding to process image boarders are the stride in which they are applied are defined by the layers. The feature map is the output from convolutional layers and a convolutional layer with *n* filters produces *n* feature maps, which become the next layer input. For backpropagation, the gradient of the convolution is needed which is the forward-pass convolution with weights flipped along every axis.

Pooling layer is applied for dimensionality reduction in convolutional neural network. The main purpose of pooling layer is to eliminate the unnecessary information and only remain the significant and critical information. As the data pass through the network, pooling layer which is non-learnable layers, reduces the spatial dimensions of the feature maps with the association of some kernel of size $k \times k$ and a strides. The pooling layers are normally categorized into two classes which are average pooling layers and maximum pooling layers. The average pooling function performs an average at each position of the kernel such as a normal convolution with the kernel values all set to $\frac{1}{k^2}$. The maximum pooling layer performs a max operation with the

elements of the feature map at each position of the kernel and eliminate the information of the non-maximum neurons.

Fully Connected Layer represents a layer in which the output is fully connected with the previous layer. These are typically used in the last stage of CNNs to connect the output layer and construct the desired number of outputs. It is commonly added to CNNs to perform classification on the features extracted by the convolution and pooling layers. The output vector is then passed into a softmax function for classification scores (Girshick *et al.,* 2016).

Transfer learning is transferring learned features of pre-trained network to a new detection case. It is possible to fine-tune all the layers of the pre-trained network or fix the initial layers of the network which have more generic features like edge or color, and only fine-tune the last few layers which are representing higher-level portion of the network to learn specific features of the new dataset (typically a smaller dataset). Compared to training a new CNN, transfer learning usually needs a lesser training time.

Traditional HOG (Dalal and Triggs, 2005), SVM (Cortes *et al.*, 1995) and Deformable Part-based Model (DPM) (Felzenszwalb *et al.,* 2010) use slide window approach where the classifier is processed at evenly speed locations throughout the entire image. However, Convolutional Neural Network (CNN) works differently. Girshick *et al.* (2016) proposed a regional-based convolutional neural network (R-CNN), which is a general object detection strategy that combines regional proposal and convolutional neural network (Krizhevsky *et al.*, 2012). R-CNN extract potential bounding boxes using regional proposal methods like Selective Search (SS) and then classified the bounding boxes using CNN-based classifier. Post-processing is applied to refine the bounding boxes, reduce the duplicate detections and rescore the boxes based on other objects in the scene after classification. Nevertheless, training a R-CNN

model is expensive in term of memory and time usage. Therefore, by sharing computation of convolutional layers among region proposals for an image to replace Selective Search (SS) with a neural network which is Region Proposal Network (RPN), Fast R-CNN (Girshick, 2015) and Faster R-CNN (Ren *et al.*, 2017) can reach better accuracy and lower latencies overall to a single input scale.

Instead of having a sequential pipeline for object classification and region proposals, one-state detectors regresses the coordinates of objects using end-to-end framework. Redmon *et al.* (2015) proposed You Only Look Once (YOLO) which contains formulated object detection as a single regression problem, straights from image pixels to bounding box coordinates and class probabilities. In this method, a single convolutional network predicts coordinates of multiple bounding boxes and class probabilities for the boxes concurrently. This leads to lower latency because YOLO trains on full images and directly optimizes detection performance. Redmon et al. also introduced YOLO9000 (Redmon and Farhadi, 2017) and YOLOv3 (Redmon and Farhadi, 2018) which are improved version of YOLO in term of accuracy, small object detection and speed.

In 2016, Li *et al.* (2016) proposed a new method known as Stereo-Proposal based Fast R-CNN (SP-FRCN) for cyclists detection. Li *et al.* (2017) presented a unified framework for simultaneous cyclists and pedestrian detection, which consists a novel detection proposal method to provide a set of object candidates, a discriminative deep model according Fast R-CNN for localization and classification and a specific postprocessing step to further improve the performance of detection. This approach requires large amount of data for training even though CNN based detection methods do not need manually selected features and a classifier running all over the images.

2.3.2.1 You Only Look Once (YOLO)

You Only Look Once (YOLO) is different from prior CNN-based techniques like R-CNN, Fast R-CNN and Faster R-CNN. Instead of separating the regional generation and regional classification, YOLO applies single network that takes one image as input and predicts the bounding box location as output. This detection method proposes object detection as a regression problem to spatially associated class probabilities and separated bounding boxes. This one-state detector directly regresses bounding boxes of objects in an end-to-end framework. Since the bounding boxed and class probabilities are predicted directly from whole images in one evaluation, the detection performance can be optimized end-to-end directly compared to region proposals method.

YOLO is a CNNs model with unified detection can detect multiple objects in an image concurently through a single neural network in a regression formulation. Input image is divided into $S \times S$ grid and each grid cell predicts the bounding boxes of objects, confidence scores and class probabilities. For each grid cell, it predicts B bounding boxes and each box has one confidence scores. The confidence score of each bounding box show the probability that the bounding boxes contains an object. Every grid cell only detects one object regardless of the boxes number, B. Every bounding box contains 5 elements (*x, y, w, h, score*) which are centre *x*, centre *y*, width, height and confidence score. How probably the box has an object and how precise is the boundary box are reflected by the score. The width *w* and height *h* of the bounding box are normalized by the image width and image height. Centre *x* and centre *y* are offsets to the corresponding cell. Conditional class probabilities, *C* which means the probability that the detected object comes from a particular class is predicted from every

grid cell. Therefore, the predictions are encoded as an $S \times S \times (B \times \mathbf{5} + C)$. Figure 2.2 shows that YOLO makes $S \times S$ prediction with B boundary boxes.

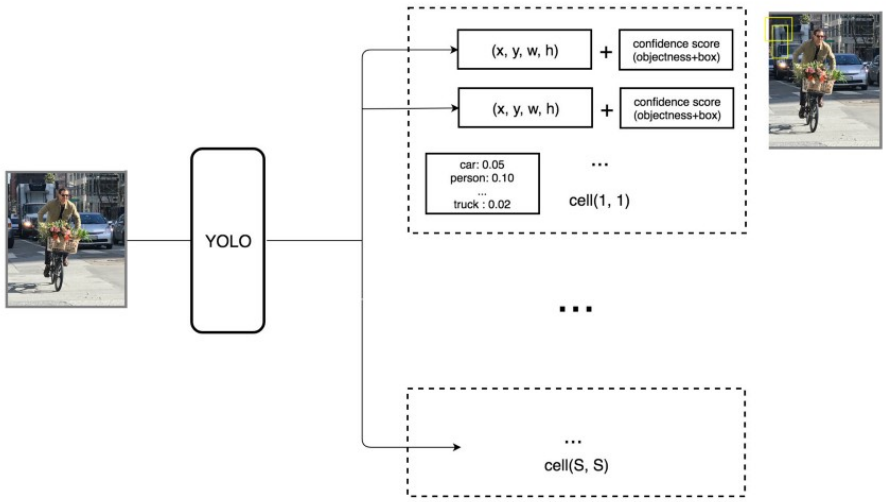

Figure 2.2: YOLO makes $S \times S$ prediction with B boundary boxes (Redmon *et al.*, 2015).

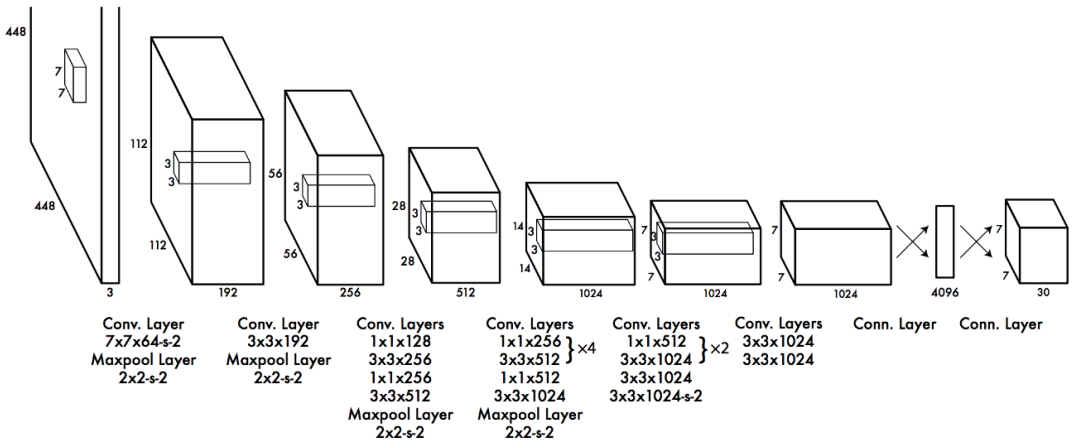

Figure 2.3: YOLO neural network architecture.

Figure 2.3 shows YOLO neural network architecture. YOLO contains 24 convolutional layers and then 2 fully connect layers. The initial convolutional layers extract features from the image meanwhile the fully connected layers forecast the

output probabilities and coordinates. YOLO uses 1×1 pixels reduction layer to minimize the depth of the features map and followed by 3×3 pixels convolutional layers. The input resolution of the detection network is 448×448 pixels and the last convolution layer outputs a tensor with 7×7×1024.

In order to process the loss for the true positive and get better at certain sizes and aspect ratios prediction, the object which contains the highest intersection over union (IOU) with the ground truth is selected among the predictions from the bounding box. YOLO directly regresses on the entire image. It applies sum-squared error among the bounding box locations and ground truth to compute the loss function. The loss function composes of:

1) Localization loss according to the bounding box center *x*, center *y*, square root of width and height from the bounding boxes.
2) Confidence loss of predicted objects unpredicted objects.
3) Classification loss in the difference of class probabilities.

The localization loss represents the deviation of the predicted boundary box sizes and locations. In order to differentiate the absolute errors in large boxes and small boxes, Square root of the bounding box width and height is applied instead of width and height. λ_{coord}, which contains default value of 5, is a scaling factor on the bounding box coordinates to ensure bounding box penalties and class probability penalties contribute equally to the loss. Confidence loss measures the objective of the box. Since most boxes do not contain objects, this loss is weighted down by a factor λ_{noobj} to penalize object identification when there is no object with default value of 0.5. The classification loss at each cell is the squared error of the class conditional probabilities for every class if an object is detected.

The final loss combines localization, confidence and classification losses together. Equation 2.1 shows the final YOLO loss function.

$$
\begin{aligned}
Final\ Loss = \ & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
& + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 \right. \\
& \left. + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(C_i - \hat{C}_i \right)^2 \\
& + \lambda_{noord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} \left(C_i - \hat{C}_i \right)^2 \\
& + \sum_{i=0}^{S^2} \mathbb{1}_{ij}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
\tag{2.1}
$$

Non-maximal suppression is applied on the detection to remove duplications with lower confidence. Overall, YOLO is well for instant processing. The detections are processed from single network therefore it can be trained end-to-end to increase accuracy. It detects one object per grid cell and enforces spatial diversity in predictions.

2.3.2.2 YOLOv2

YOLOv2 is an improved version of YOLOv1 with the goal of increasing the accuracy significantly while improving it speed. The new concepts added to idea of YOLO to improve its performance which are mentioned below:

- **Batch Normalization**

  The performance of convolutional layers is improved when batch normalization is added. This helps to regularize the model, reduce the need for dropout.

- **High-resolution classifier**

  All state-of-the-art detection methodologies apply classifier which is pre-trained on ImageNet dataset. YOLOv2 begins with 224×224 images pixels for classifier training and return the classifier with 448×448 pixels resolution using few epochs. This makes the detector training easier and work better on higher resolution inputs.

- **Convolutional with Anchor Boxes**

  Fully connected layers are removed but 5 anchor boxes are applied in YOLOv2 to predict bounding boxes. A pooling layer is eliminated so that the output of convolutional layers of network contains higher resolution. The size of input image is changed from 448×448 pixels to 416×416 pixels and the size of the output feature map is changed from 14×14 pixels to 13×13 pixels. Convolutional layers in YOLO down-sample the image by factor of 32. This creates an odd number spatial dimension and it is more certain on where the object locates. YOLOv2 predicts class and objectness for every anchor box.

- **Dimension Clusters**

  The boundary boxes have strong patterns in many problem domains. Instead of choosing anchor box dimensions manually, the top-*K* boundary that contains the great coverage of training data can be