| | |
|---|---|
| Title | 交通監視システムのための密度を意識した注意ネットワークを用いた車両密度推定 |
| Author(s) | SOOKSATRA, Sorn |
| Citation | |
| Issue Date | 2021-03 |
| Type | Thesis or Dissertation |
| Text version | ETD |
| URL | http://hdl.handle.net/10119/17478 |
| Rights | |
| Description | Supervisor:吉高　淳夫，先端科学技術研究科，博士 |

Doctoral Dissertation

# Vehicle Density Estimation Using Density-Aware Attention Network for Traffic Surveillance System

Sorn SOOKSATRA

*Supervisor:* Assoc. Prof. Atsuo Yoshitaka

Graduate school of Advanced Science and Technology
Japan Advanced Institute of Science and Technology
[Information Science]
March, 2021

# Abstract

The surveillance system is widely deployed in several applications for observing the characteristic of target density because it has an advantage in low installation and maintenance costs. In a traffic surveillance system, this characteristic is usually applied in a traffic light control system (TLCS) from the number of vehicles. It helps to control the period of traffic lights and prevent any traffic accidents from vehicles. Besides, the traffic characteristic is useful information for road users to avoid crowded regions, including vehicles, people, and others. Since this system utilizes computer vision-based techniques, it is sensitive to outdoor environments (e.g. lighting conditions, traffic viewpoints, and so on). In addition, hardware with low performance usually is deployed in the surveillance system. It means that the system should be compact and requires low computational cost for operation. Therefore, density estimation cannot be operated properly under any circumstances without concerning these conditions.

Recently, there are several related studies on density estimation. It is well known that density is originally estimated by detecting and counting their regions in the input image. The counting target regions were classified by their visual features (appearance and motion features). Instead of relying on a detection-based approach, recent studies focused on end-to-end learning methods relied on regression-based approaches. These approaches concern holistic features which are visual features extracted from whole input images predicted by a prediction model. The holistic features are utilized for mapping dense regions into density maps which represent the change of object density, where their ground truth was prepared by a convolution between Gaussian distribution and target coordinates corresponding to the target sizes. Nowadays, the task of vehicle counting via a regression-based approach is riddled with a scale-aware model to handle scaling problems. Prediction models are usually designed by several stacked CNNs in parallel, which are called multi-column network architectures. In short, one stack of CNN is used to predict density maps on a specific size of the vehicle. However, these techniques have problems related to practical application as follows:

- With the variation of camera viewpoints, the target sizes can be calculated accurately resulting in the misclassification for density map formulation in the ground truth of an estimation network.
- Recent regression models with multi-column network architectures have a large number of model parameters and high computational costs using several network architectures for various target sizes.

The purpose of this study is to examine the only one stack of CNN, which is called a single-column network architecture, and reduce computational costs and model parameters while keeping similar counting accuracy to multi-column network architectures. It also should not suffer from a scaling problem by avoiding the utilization of target sizes. It found that the target size can be categorized by their local densities which are related to

distances of their neighbor target for preparing the ground truth. The proposed prediction model chooses to investigate the connections between feature maps from different layers, where holistic features of small and large objects can be extracted from feature maps in shallow and deep layers, respectively. The connection can be done by skip connections to integrate feature maps from shallow layers with another in deeper layers. This process, which is called forward connections, can recover the holistic features of small objects. On the other hand, the backward connections are introduced by extracting feature maps from deep layers to combine with the shallower layer. It can be expected that information on a deep layer can help to optimize the shallow layer, where the performance in an earlier stage can be improved. Considering the quality of a density map, feature maps in every layer should have the same resolution to prevent information loss from adjusting their resolutions. Then, pooling layers are replaced with a dilated convolutional layer. Therefore, the contributions of this dissertation can be summarized as follows:

- Instead of relying on the target size, the proposed density map utilizes average distances among target samples where it is designed for visualizing the difference pattern of various vehicle densities (high and low density regions).
- To reduce computation cost, a single column network architecture for vehicle density estimation is designed by including skip connections and dilated convolutions to extract holistic features from intermediate convolutional layers and keep semantic information for density map estimation, respectively.

Since vehicle density estimation is mainly focused on this research, all models are evaluated by the common criteria for vehicle density accuracy. The state-of-the-art of regression-based approach was implemented for comparison. In addition, well-known CNNs with skip connections (e.g. U-net, Resnet, and Densenet) were also applied for analysis. The empirical results show that the proposed prediction model with backward connections achieved a vehicle density accuracy 92.47 % which is close to accuracy of state-of-the-art (93.33 %). From this result, the achievement is summarized as follows:

- In the density map configuration, the target size estimated by an average distance of the target is insensitive to camera viewpoints.
- From the point of view of counting accuracy, the proposed method with a single-column network achieves a promising result which is close to the vehicle counting accuracy from a multi-column network or network with a large number of model parameters.

Moreover, the proposed network satisfies with the minimum requirement of TLCS and it is effective to reduce under-counting errors. However, there is a room for improvement in other datasets consisting of overlapping target regions or crowd counting datasets.

**Keyword**: surveillance system, vehicle density estimation, regression model, skip connection, dilated convolution, traffic analysis.

# Acknowledgments

I wish to express his sincere gratitude to his supervisor Associate Professor Atsuo Yoshitaka at Japan Advanced Institute of Science and Technology. I appreciate his broad and deep knowledge and patience whenever we discuss. He provided the valuable comments information and point out my mistakes in my study. Not only computer vision, his guidance help me to find connection of the practical applications in other fields, where a research goal can drawn. Without his supervision, there seems to be no end in sight to my PhD course.

I also sincerely like to extend my heartfelt gratitude to my thesis advisor, Assoc. Prof. Toshiaki Kondo at Sirindhorn International Institute of Technology who supports me since the beginning of my Phd program. His advice gives me ideas and concepts for further analysis in my research, He also help me to publish journals and conference papers. Without his supervision, my work seem to be unseen.

I also sincerely like to thank Dr. Pished Bunnun, my co-advisor, at Thailand's National Electronics and Computer Technology. He supported me hardware and software for my experiments. He shared me his experiences to improve my skill. Without his supervision, my vision seem to be underdeveloped.

I greatly appreciate my classmates and my fellow doctoral students in JAIST Dual Degree program for all the fun we had while we studied. I am also grateful for my friends in Yoshitaka's Lab at JAIST to give feedbacks of my simulation, coding, and my research problems.

Last but not least, I would like to appreciate my family who supported me spiritually throughout my life especially my mother who always take care of my health while I was working. She always understands and supports me everything in all situation, consoles me when I feel despair. All of the words that mean thank you is not enough to describe my feeling.

This research is financially supported by The Ministry of International Affairs and Communications of Japan, Japan Advanced Institute of Science and Technology, Sirindhorn International Institute of Technology (SIIT), and Thammasat University (TU).

# Contents

This dissertation was prepared according to the curriculum for the Collaborative Education Program organized by Japan Advanced Institute of Science and Technology and Sirindhron International Institute of Science and Technology, Thammasat University.

# List of Figures

# List of Tables

# Chapter 1

# Introduction

In recent years, the number of vehicles is exponentially increased, especially in urban areas. The traffic on the road has become complicated, causing traffic congestion and car accidents. It reveals that an enormous amount of time and money is wasted (e.g. 5.5 billion hours of time delay and 2.9 billion gallons of fuel wastes in urban areas) due to traffic congestion from 2000 to 2010 [47]. It predicts that the congestion cost will increase from $121 billion (in 2011) to $199 billion (in 2020). A Traffic Control System (TLCS) typically utilizes static the interval of traffic lights at intersections and does not provide priority to emergency vehicles such as ambulances, firefighters, and police cars. It possibly causes a loss of lives, damage or destruction of property, and increase fuel costs, pollution, and congestion. To improve system performance, the interval of traffic lights should be dynamically adjusted according to a traffic attribute in each location. Therefore, the traffic attribute is the main factor in traffic management.

A typical Traffic Light Control System is depicted in Figure 1.1, consisting of four functions (e.g. information collection, data diffusion, required activities planning, and suitable action implementation). To carry out these functions, a Road Side Unit (RSU) with several sensors and a Traffic Management Centre (TMC) is located at the corner and center of the intersection as shown in Figure 1.1. These sensors collect the real-time traffic attributes (e.g. vehicle density, type of vehicle, average waiting time, and pollution) and relay the traffic attributes to RSU. Finally, RSUs gather all information and send it to TMC for evaluating the proper green and red traffic light intervals. This research focuses on vehicle density estimation for the data collection system in TLCS mentioned above.

It is well known that a traditional inductive loop detector is replaced with a surveillance camera at signalized intersections [32] because it provides a low cost for installation and maintenance. The TLCSs with a surveillance camera is known as Video Imaging Vehicle Detection Systems (VIVDS). A typical VIVDS consists of three key components: one or more video cameras, a central image processor, and detection software. Surveillance cameras are utilized for recording a video at each junction. A central image processor unit analyzes the video signals from the cameras. Since programmable detection zones and detectors are set up in the central image processor, the detection zones and detectors are activated when vehicles pass the detection zones or detectors. The performance of VIVDS is depended on visual features and various conditions of traffic videos (e.g. camera viewpoints, lighting conditions, weather, and so on). The visual feature represents a characteristic of objects (e.g. shape, texture, color, and so on). The errors in VIVDS are categorized into false detection and missed detection. A false detection occurs when a


Figure 1.1: A typical Traffic Light Control System [58]

detector is activated by vehicles in adjacent lanes, vehicle shadow, the shadow of buildings or trees, and abnormal driving. False detection usually results in more counts than the true counts in the field. A missed detection occurs when a detector is not properly triggered while a vehicle passes through the detector.

## 1.1 Vehicle density estimation using computer vision

As mentioned in the previous section, vehicle density estimation using a surveillance video relies on a visual feature in computer vision. These features are utilized to extract vehicle presence on the road. The visual features are categorized into two types: appearance-based and motion-based features. Similar to the human visual system, vehicles are identified from their appearances, compared with other objects in the traffic images. Appearance features represent the vehicle characteristic, for example, class, sizes, shapes, and colors. These attributes are employed as prior knowledge to segment foregrounds (containing the objects of interest) from backgrounds (its complementary set). Several appearance features were utilized as vehicle cues (e.g. symmetry, color, edge (horizontal/vertical), shadow, and so on). At daytime, as shown in figure 1.2a, shapes of vehicles are salient where they can be extracted to detect vehicles. However, the features mentioned above are hardly identified in the nighttime as shown in figure 1.2b. The only salient visual features are headlights, rear lights, and beams. Since appearance-based features currently suffer from lighting conditions, a problem can be solved by using a specific detector in various illuminations. Appearance features are deployed in the daytime, while a night-time detector utilizes a headlight as the main feature. Therefore, many studies only focus on one or two lighting conditions.

Since vehicles are moving objects in traffic videos, motion-based features can be extracted for vehicle density estimation, where regions are considered as foreground (moving

(a) (b)

Figure 1.2: Examples of traffic images in (a) daytime and (b) night time [89].

objects) and background. These features are deployed to distinguish between the foreground and background in images. Unlike appearance-based features, the motion-based feature employs temporal information or frame sequence to detect moving objects. Moving regions can be extracted by finding the difference between two consecutive frames. Since motion-based features are usually deployed without any prior knowledge, they are insensitive to vehicle appearances in various conditions of traffic videos. However, they are sensitive to noises in the complex background and lighting conditions in traffic videos. In addition, objects with low and high speed cannot be archived by this technique.

The schematic flow of vehicle density estimation is depicted in Figure 1.3. After vehicle features were extracted in the first step, the classifier has a role to distinguish between vehicle and non-vehicle objects in images. Classifiers learn the characteristics of vehicle features statistically and learn a decision boundary between the vehicle and non-vehicle objects. To optimize appearance-based classification performance, huge inter-class variability should be extracted. It is well known that Convolution Neural Network (CNN) achieves promising object recognition and classification performance [93]. Most recent studies deploy deep learning techniques based on CNN for vehicle identification and detection. In addition, CNN can be utilized for extracting features and classifying objects at the same time. However, a large amount of data is required for deep learning techniques compared with a statistical approach.

In the last step, vehicle regions obtained from classifiers are counted. The vehicle counting approach is categorized into two types [73], i.e., detection-based and regression-

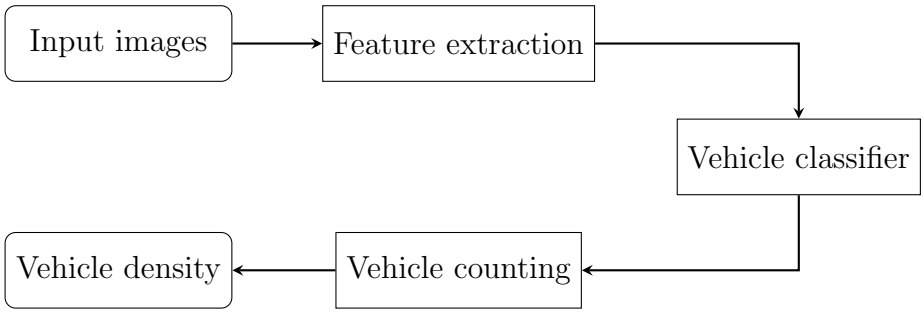



Figure 1.3: Schematic flow of vehicle density estimation.


Figure 1.4: Example of detection-based approaches, where blue and red bounding boxes are represented as vehicle locations in the images [88].

based approaches. The detection-based approach focuses on visual features of the vehicle in classifiers, where a sliding window detector is used to detect vehicles in the scene. The main goal of detection-based approaches is to extract vehicle regions in an image. As a result, vehicle regions are represented as bounding boxes for localization and tracking as shown in Figure 1.4. It means that the number of bounding boxes is calculated as vehicle density. In addition, the results from this approach can be deployed for other applications for traffic analysis (e.g. vehicle type identification, speed measurement, and so on). Even though various applications rely on detection-based approaches, its performance suffers from traffic images with high vehicle density because vehicle appearance is occluded from other objects or vehicles.

Since the detection-based approach was not successful for vehicle counting in the presence of the extremely dense crowd and high background clutter, this issue was solved by using a regression-based approach. This approach learns a linear mapping between holistic features and corresponding objects to generate density maps, thereby incorporating spatial information in the learning process. Unlike detection-based approaches, localization and tracking are ignored by applying an algorithm for estimating image density whose integral over any region in the density map giving the count of objects within that region as shown in Figure 1.5. The regression-based approach has two main components as follows:

- **Density map** shows the spatial distribution of vehicles in specific regions, where they are formulated from holistic features in the regression approach. Gaussian process regression is usually employed to compute a density map from low-level features.

- **Prediction model** predicts the density map from input images. This model can be designed with various learning algorithms (e.g. random forest regression, decision

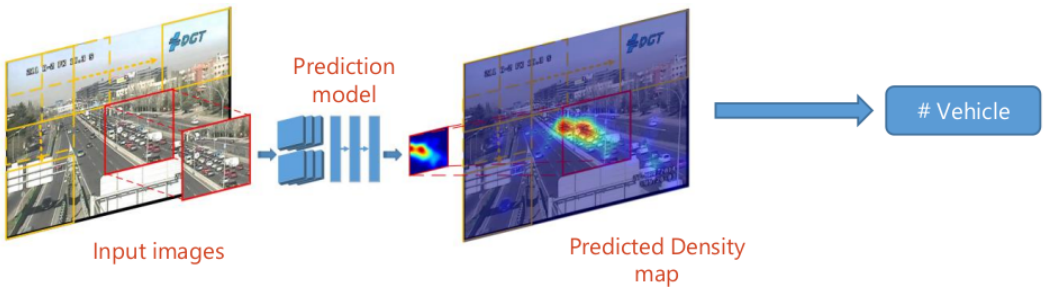



Figure 1.5: Schematic flow of regression-based approaches for vehicle counting [61].

tree, and so on). However, CNNs are usually deployed in recent studies because of their success in numerous computer vision tasks.

In the recent studies of traffic analysis, both detection-based and regression-based approaches were studied to deploy their specific application. Since various traffic density is evaluated in VIVDS, especially in traffic images with high vehicle density, regression-based approaches are mainly focused in this research.

## 1.2 Traffic video conditions

Since visual features depend on the resolution and contrast of traffic videos, the image quality becomes one of the main factors for vehicle density estimation. The specification of a traffic surveillance camera is based on regulation in different areas. Unfortunately, traffic video quality typically has low resolution and low frame rate to reduce power consumption and installation/maintenance cost [51]. In addition, color information is not available in some areas. The typical specification [21] of traffic videos can be summarized as follows:

- The video compression (MPEG4, H261, H263, H264, and H265)
- Resolution ($176 \times 120$, $352 \times 240$, $720 \times 480$)
- Frame rate (1 to 30 frame per second)

Figure 1.6 reports a survey of the average frame rate in traffic surveillance. Since the typical video frame rate is around 25 - 30 frames/second (fps), most traffic videos (with frame 6 -10 fps) cannot accurately estimate motion features because motion trajectory is not visible in the video with low fps. By contrast, the video resolution has increased by $\sim 10x$ that of frame rate, showing the importance of more resolution vs fps in video surveillance. Therefore, appearance-based features should be deployed in VIVDS to handle practical circumstances.

Since the traffic videos are recorded in an outdoor environment, image quality depends on its environments (illumination and weather). Illumination is divided into daytime and nighttime as shown in Figure 1.2. They affect image intensity and contrast which help to visualize the texture or appearance of vehicles in a video. The illumination is sensitive to

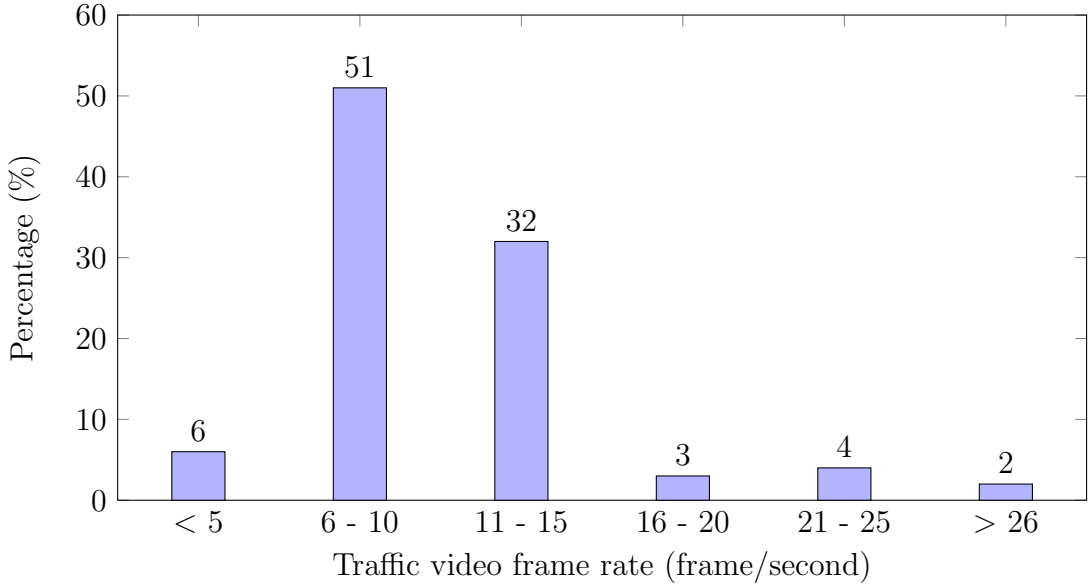


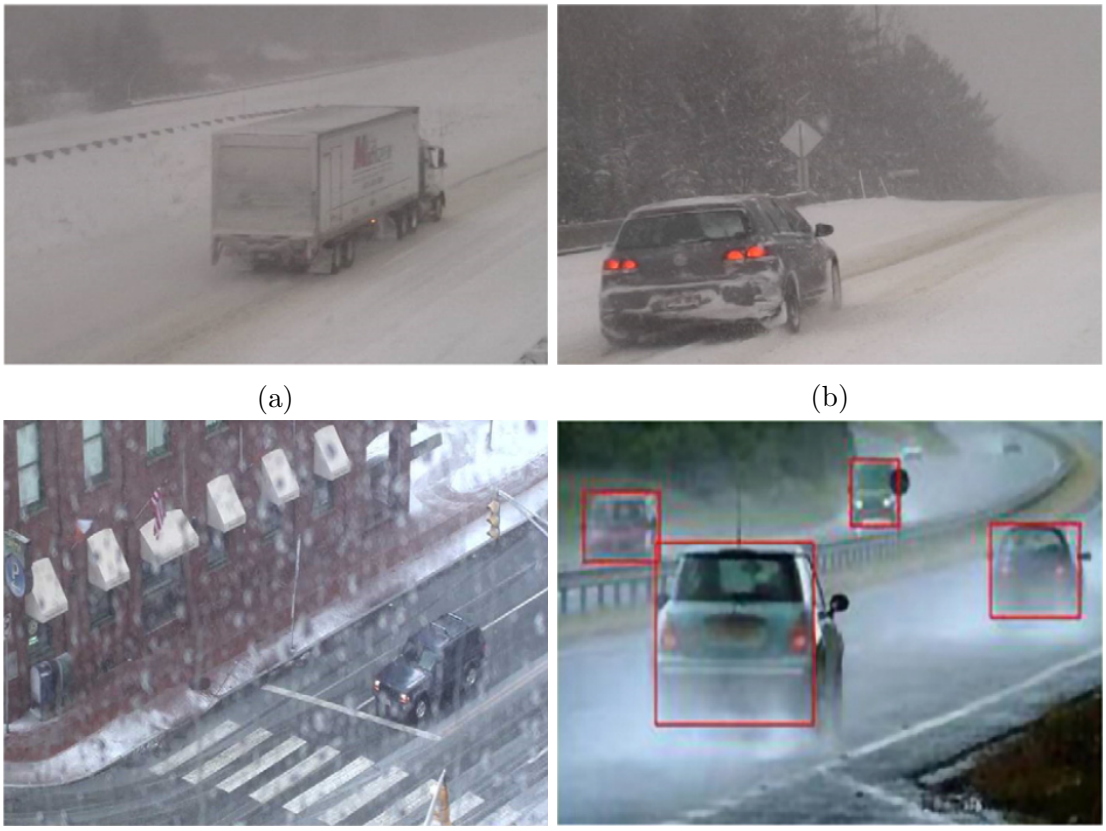

Figure 1.6: Average frame rate usage [30]

the natural light source (e,g, sun, moon, and so on) as summarized in Table 1.1. At the point when traffic cameras are deployed in a dark environment (i.e., 0 Lux), vehicles can not be captured by cameras. In this condition, artificial light sources (e.g. street light, headlight sources, and so on) help illuminating traffic images. However, these lights can affect the color cues in the traffic images.

On the other hand, the weather can be divided into normal and adverse weather. Normal weather is the best condition of vehicle density estimation in VIVDS, by giving a promising performance of vehicle counting and detection. In practical cases, the vehicle is difficultly counted in adverse weather because of high occlusion, vagueness, and blurring. Figure 1.7 illustrates examples of traffic videos with adverse weather. a traffic image with a blizzard is ambiguous between background and foreground as shown in Figure 1.7a. an appearance feature can not extracted when a vehicle is covered with snow in Figures 1.7b and 1.7c. In addition, rain or water blurs images in Figure 1.7d.

| Time period | Conditions | Luminance intensity (in Lux) |
|---|---|---|
| Daytime | Sunny | 10,000 - 1,000,000 |
| Daytime | Cloudy | 100 - 10,000 |
| Daytime | Dawn twilight | 1 - 10 |
| Nighttime | Full moon over head | 0.1 - 1 |
| Nighttime | Quarter moon | 0.01 - 0.1 |
| Nighttime | Sunny starlight | 0.001 - 0.01 |
| Nighttime | Cloudy starlight | 0.0001 - 0.001 |

Table 1.1: Various outdoor illumination conditions at different places [63].



(a) (b) (c) (d)

Figure 1.7: Examples of traffic images with adverse weather in (a) blizzard, (b) snow fall, (c) wet snow, and (d) the rain [87].

## 1.3 Motivations and Research goal

As described in previous sections, the vehicle density estimation method should count the number of the vehicle in various vehicle density, to handle the exponential growth of vehicle. Since a detection-based approach suffers in high vehicle density, a regression-based approach is concerned in this research. However, their recent studies deploying prediction models require a large number of training parameters and memory space because of several CNNs called multi-column network architecture. one stack of CNN was deployed for handling various vehicle density, calculated from vehicle sizes. These conditions are not suitable in VIVDS, mentioned in section 1.2, which deploys videos with low frame rate and low power consumption. Therefore, there are drawbacks of these approaches which are summarized as follows:

- Actual vehicle sizes cannot be obtained due to occlusion in many cases, causing an error in density map estimation.
- Due to a large number of the model parameter, the traffic density might not be reported within a time requirement in VIVDS by using a multi-column network.

My objective of this study is to examine the single-column network requiring only one CNN in a prediction model. The advantage of this approach is the reduction of computational costs and the number of model parameters. The proposed method chooses

to investigate intermediate layers of CNN whose features generate a density map of the vehicle in a different size, by using skip-connection techniques. To minimize semantic information loss in deeper layers, dilated convolutional layers are applied in this research. Without referring to vehicle size, vehicles are categorized by average distances to their neighbor vehicles described in Chapter 3. Therefore, the research contributions are summarized as follows:

- Density map by geometry-adaptive kernels is proposed by calculating average distances among target samples where it is designed for solving an occlusion problem and local density is taken into account.
- Without implementing several CNNs, a single-column network is enhanced by using skip connections and dilated convolutions to increase counting accuracy and density map quality.

From the above statement, the significance of this study is stated that the promising performance for object counting can be achieved by the single-column network, where it can be further developed as a backbone network in future work. With a compact size of the proposed backbone model, it can minimize the hardware requirement for further development. In addition, the proposed method can be applied in other datasets because a supervised learning-based technique was applied.

## 1.4 Structure of the Dissertation

The remaining parts of this thesis are organized as follows:

- **Chapter 2** shows a review of related studies of object counting in the detection and regression-based approaches. The literature review mainly focuses on recent studies with CNN by visualizing and analyzing their network architectures in different circumstances. Finally, the problem of size estimation in the multi-column model is defined and find possible solutions.
- **Chapter 3** defines the requirements and conditions of input data related to the proposed method. density map formulation is described by proposing the density map by geometry-adaptive kernels. Limitations of traditional density maps are mentioned. It also covers the comparison between traditional and proposed density maps where they are evaluated in various vehicle density.
- **Chapter 4** explains contributions and illustrates the architecture of the prediction model visually. The network configuration is defined in detail (e.g. receptive field sizes, number of parameters, and so on) which affects by the time or memory consumption of the system. Since the number of training images is insufficient, the training process is described to overcome this issue.
- **Chapter 5** evaluate and report the performance of the proposed density map and prediction model. The common criteria of density estimation evaluation are defined and it shows the minimum requirement of VIVDS. The advantages and drawbacks are analyzed in different conditions and aspects, compared with existing methods,

related to detection and regression-based approaches. Since the previous studies are evaluated by their own criteria, their performance is converted to common criteria for comparison.

- **Chapter 6** summarizes knowledge studied from this research. It describes the achievements of the experimental results in a previous section and discusses the proposed methods relevance to the existing crowd and vehicle density estimation methods. Besides, drawbacks are identified and discussed to give direction for future research.

## 1.5 Summary

In this chapter, vehicle density estimation related to VIVDS was discussed. To minimize installation and maintenance costs, computer vision-based techniques were deployed in VIVDS. The techniques for vehicle density estimation are categorized into detection-based and regression-based approaches by using the appearance and motion features of vehicles. Even though computer vision-based techniques are usually employed, they are sensitive to traffic video conditions (e.g. lighting conditions, camera viewpoints, and so on). In addition, the RSU is typically compact and low system specification providing low frame rates and image resolution. Therefore, the primary goal is to minimize the number of model parameters and computational cost for vehicle density estimation by focusing on a single-column network, while this network has similar performance to a multi-column network.

# Chapter 2

# Literature Review

There is a vast amount of literature on vehicle density estimation. Since the recent study usually focuses on the change in vehicle density, object counting is taken into account in their works. Object counting based on computer vision techniques is primarily depended on vehicle detection or localization, where the number of the target is obtained from the target region as mentioned in the previous chapter. With the exponential growth of vehicles, vehicle detection suffers from an occlusion problem in traffic videos with high vehicle density or crowd images. To solve this issue, traffic images are regressed to low-level images, called a density map, which is formulated to the number of vehicles. Therefore, vehicle density estimation is categorized into the detection and regression-based approaches, described the following sections:

## 2.1 Detection-based approach

Initial works in this field focused primarily on the detection style framework, where a sliding window detector is used to detect object in the image. This information is utilized for counting the number of the object. Vehicle detection is an approach for extracting vehicle regions (foreground) from background regions. These regions are classified by visual features obtained by the following methods:

### 2.1.1 Appearance-based methods

Same as the human visual system, vehicles are recognized from their appearances, compared with other objects in the traffic videos. Due to a limitation of memory and time consumption in VIVDS, low-level features (e.g. color, shadow, symmetry, and edge) are typically extracted for localizing vehicle regions. These features are simple, efficient, and of great usefulness in vehicle-related information extraction which is described below.

- **Color** provides rich information in visual features because vehicle regions usually have higher color saturation than background regions (roads). This information is usually extracted in various detection application related to vehicle counting (e.g. vehicle lights [14] in night time and license plates [1]). Since a traffic video is sequence of RGB or grayscale images which are sensitive to lighting conditions, their images are converted into different color model (HSV [60, 62], YCBCr [86, 15], CIELab [10]) to reduce the effects of lighting conditions.

- **Shadow** is often detected along with a vehicle, where it is critical for improving the accuracy of vehicle detection and tracking. Shadows are typically classified by their shapes and colors. The shape of a shadow is classified by the shape of the object and lighting conditions, which can be exploited to detect the shadow [94]. However, a vehicle shadow may cause many problems such as merging, shape distortion and a loss of objects due to illumination changes. On the other hand, color information typically utilizes color models based on contrast [5], brightness [31], mean and variance of all color components [31].

- **Symmetry** is the main properties of manufactured products, including vehicles. In frontal and rear viewpoints, the image of a vehicle usually has symmetry property which is for detection cues in VIVDS. These cues usually are defined by the symmetry axis [48] which help to indicate the center lines of symmetrical objects. Then, they can be classified by symmetrical features (e.g. SIFT [49], SURF [7], and so on). However, this information is sensitive to camera viewpoints because current video cameras are installed at locations with different angles and views. Therefore, symmetry is not suitable for vehicle detection.

- **Edge** has been widely deployed in VIVDS because of low computational complexity. Traditional edge detection is employed by spatial filters such as Sobel, Canny, or Prewitt filters to generate vertical and horizontal edge maps. In a state-of-the-art of vehicle detection-based approach, multi-scale edge fusion [57] was to locate target vehicles. Edge often plays the role in provision of contour information, while other features mentioned above provide contextual information to extract vehicle candidates.

Even though low-level features are extracted fast, the appearance-based method with only one low-level feature is not able to provide promising performance and represent all useful information. Therefore, previous studies usually combine contextual and contour information. This solution extracts visual features of a vehicle in a more effective manner. On the other hand, the traditional method usually constructs a feature descriptor from those low-level features. Feature descriptors efficiently localize an object in traffic videos. In ideal cases, descriptors should deal with various objects and robust to various camera viewpoints. In addition, it is invariant to geometric transformation [44]. The feature descriptors are described as follows:

| No | Appearance feature descriptors | Camera viewpoints | Application domain |
|---|---|---|---|
| 1 | Underneath, vertical edge, symmetry [13] | Rear | Highway scene detection |
| 2 | Sketch, texture, color, flatness [45] | Front | Complex urban traffic conditions (with occlusion) |
| 3 | Color, corner and edge [80] | Top | Parking area |
| 4 | HOG and Haar-like features [83] | Rear | Freeway |
| 5 | Shadow, illumination entropy, edge [75] | Rear | Overtaking |
| 6 | Appearance & edge based feature [16] | Oblique | Blind-spot detection |

Table 2.1: A brief summary of vehicle detection approaches based on feature fusion.

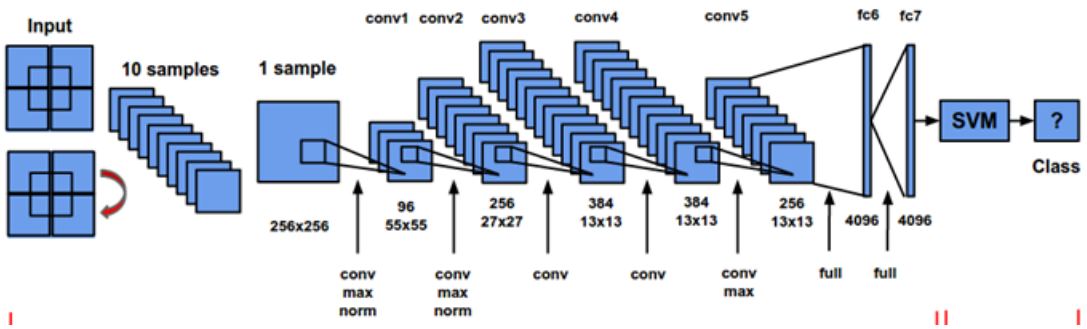



Figure 2.1: Structure of Convolutional Neural Network (CNN) [39].

- **Histogram of the gradient (HOG)** calculates the edge of gradient structure for representing a characteristic of a local shape because edge information is insensitive to camera viewpoints and lighting conditions. Several types of HOG were deployed in VIVDS. Pyramid Histograms of Oriented Gradients (PHOG) was extracted from traffic images as basic features [40]. Three different configurations involving horizontal (H-HOG), vertical (V-HOG) and concentric rectangular (CR-HOG) descriptors were proposed to perform in a more cost-efficient manner than the original HOG [4]. By employing symmetry properties, symmetry HOG vectors were developed for vehicle verification [16].

- **Haar-like feature** was first introduced by Viola and Jones [82] where it includes edge, line, and center-surround features for representing the vehicle characteritristic. This descriptor was deployed in many vehicle detection approaches. Harr-like feature extraction method was used to represent a vehicle's edges and structures [90]. A 2D triangle filter was proposed based on Haar-like features to detect vehicles [90]. Sivaraman and Trivedi [75] provided a general learning framework using Haar-like features. The computation of Haar-like features is fast. However, the dimension of this feature vector generated from images is high [83]. An operation of dimension reduction was then applied to decrease memory space.

Table 2.1 illustrates the examples of vehicle detection with various feature descriptors. It shows that their algorithms were focused on specific purposes (application domains) and camera viewpoints. As a result, they were not deployed in practical environments. In recent years, VIVDS was improved by utilizing deep learning techniques. CNN is the most popular deep learning technique for vehicle detection. It optimizes target's features from feature representation mentioned above. Input data is divided into training and testing input data, where they are trained by convolution with several image filters, called CONV in each layers, as shown in Figure 2.1. As a result, fully connected layers (fc) is classifier resulting recognition results.

Detection using CNN usually perform either in monolithic style or part-based approaches. The monolithic detection approach focuses on the feature representation of whole vehicles. Region with CNN (RCNN) [26] is currently state-of-the-art for vehicle detection in a monolithic approach. In general, CNN is deployed for classifying object classes without localization. Instead of applying sliding windows in whole images, RCNN deploys region proposal techniques to extract possible regions containing targets. Then, candidate regions were classified by CNN as shown in Figure 2.2. Region proposal tech-

niques can be done as following steps:

1. Generating initial sub-segmentation regions
2. Using greedy algorithm to recursively merge similar regions.
3. Using the generated regions to produce the final candidate region proposals.

Since RCNN is state-of-the-art in this field, several studies implemented and developed RCNN for various purposes. Geometric proposals for Faster R-CNN (GP-FRCNN) [2] rank the generic region proposals with an approximate geometric estimate of the scene. However, this simple extension requires scale adjustments (e.g., anchors, layer resolution). GP-FRCNN is effective for smaller and larger objects and does not require an explicit geometric formulation. Lightweight SSD based on ResNet10 with dilations (SSDR) [3] focused on the reduction of time computation based on Resnet 101, pruning can be done on a per-channel basis by eliminating less important channels in convolution filters. R-CNN with Sub-Classes (RCNNSC) [68] is based on training in different sub-class to improve performance accuracy. Instead of using a single object class (i.e., the vehicle) to train on the R-CNN, multiple sub-classes of vehicles (i.e., car, van, bus, and others) are used, such that the RCNN learns respective features of each vehicle class better. Faster R-CNN with ResNet101 (FRCNN-Res) [34] employs the Faster R-CNN architecture, where the ResNet-101 model is used to adjust the performance on the training set.

On the other hand, part-based detection was designed for solving occlusion problems, especially in traffic videos with high vehicle density. The most popular one in these approaches is the Deformable Part Model (DPM) [18] which constructs boosted classifiers for specific vehicle parts (e.g. windshields, doors, wheels, and so on) to estimate the vehicle counts in a designated area. The distinct parts are detected based on an appearance feature to identify and localize vehicles. DPM consists of a global root filter and part filters, represented as red and yellow circles shown in Figure 2.3, to detect and track vehicles on road using part-based transfer learning. Region-based Deformable Fully Convolutional Network (DFCN) [85] is based on Resnet-101 as a backbone, where a deformable model is combined to provide accurate localization and correct classification of objects for stationary cameras mounted near signalized traffic intersections. Vehicle detection by independent parts (VDIP) was introduced in [76] for urban driver assistance. Front, side, and rear parts were trained independently using active learning. Part matching classification using a semi-supervised approach form vehicles oblique view from independently detected parts. A rear view vehicle detection was considered in [79] based on multiple salient parts that include license plate and rear lamps. For part localization, distinctive color, texture and region features were extracted. Then Markov random field model was used to construct a probabilistic graph of the detected parts. Vehicle detection was accomplished by inferring the marginal posterior of each part using loopy belief propagation.

### 2.1.2 Motion-based methods

Since vehicles are usually moving objects in traffic videos, motion can be utilized as features. In motion features, regions can be considered as foreground (moving objects) and background. These features are used for extracting or separating moving objects from

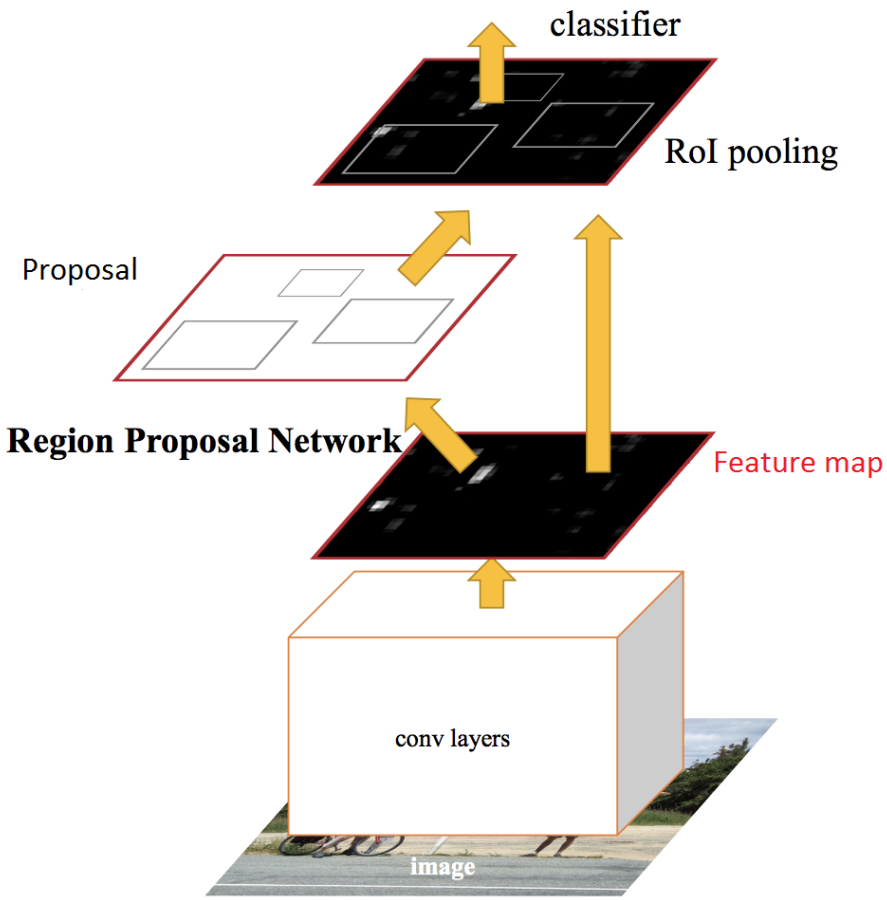



Figure 2.2: The procedure of Region with Convolutional Neural Network (RCNN) [26].

backgrounds which are static regions in images. Unlike appearance-based method, motion-based method applies temporal information or frame sequences to detect moving objects. In addition, they are insensitive to vehicle types because prior knowledge of appearance is not required. Frame difference is the most popular technique of motion-based methods because of low computational cost, where the pixel-wise difference was computed within a frame sequence. However, it is sensitive to noise and lighting conditions in a traffic video. In addition, the drawback of frame difference is that there are holes in the vehicle regions when the objects move slowly. On the other hand, vehicle trajectory is not present when the objects move fast. Background subtraction [53] is extended work from frame difference by referring a reference frame or background model which is utilized for comparison with current frames. Its flow chart is shown in Figure 2.4, where there are three types of background model as follows:

- **Parametric background modelling** is estimated by uni-modal probability density function, and update the distributions parameters. Frame averaging [37] is conventional by averaging intensity within a frame sequence from the summation of $N$ frames within a frame sequence. To improve its performance, Gaussian filter [92], median filter, and sigma-delta estimation [54] were added. There is a remaining challenge for parametric background modeling such as slow or temporary stopped vehicles, lighting conditions, and traffic images with high vehicle density.

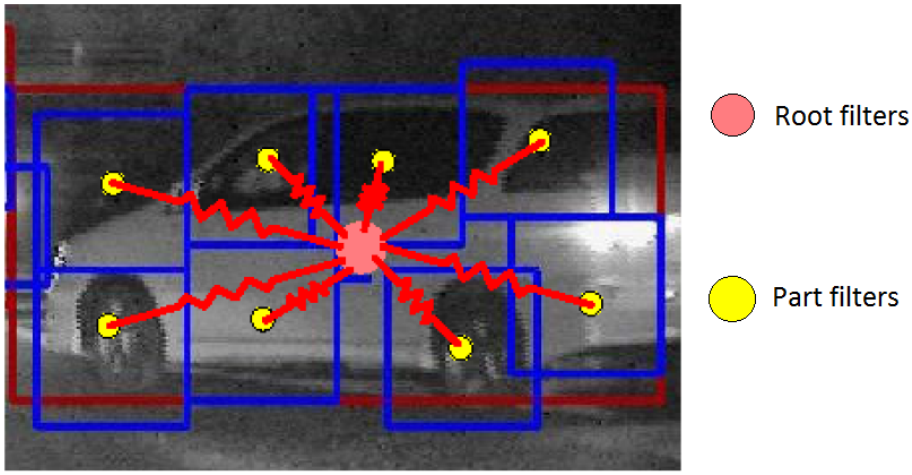



Figure 2.3: Part-based detection models where red and yellow circles are represented as root and part filters, respectively [76].

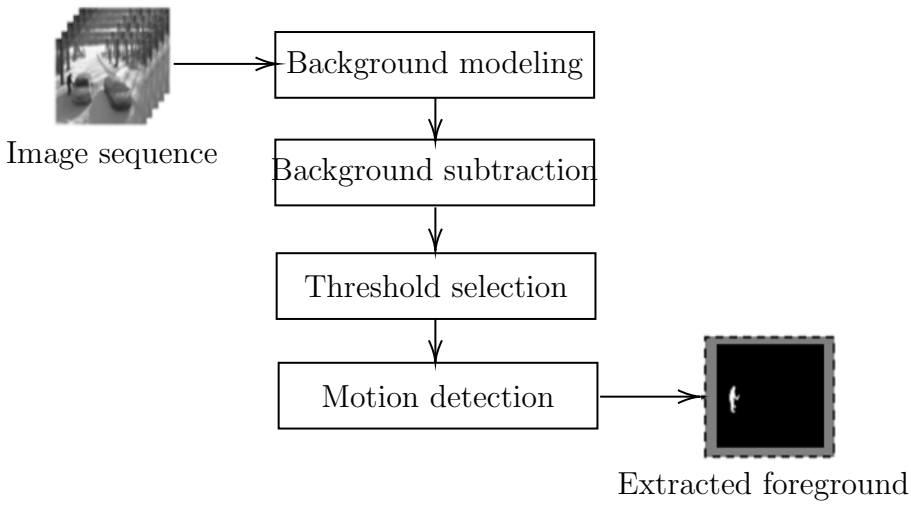



Figure 2.4: Schematic flow of moving object detection by background subtraction [29].

- **Non-parametric background modeling** keeps a sample of intensity values for each pixel in the input images, where probabilistic representation is estimated. Based on recent studies, the examples method of this techniques are Kernel Density Estimation (KDE) [20] and codebook [41]. These techniques were adapted with rapid changes in the environment resulting in high detection sensitivity. However, it requires space for storing samples of intensity from frame sequences. Therefore, it is not suitable for the long-time application.

- **Predictive background modeling** is employed in modeling the change of pixel intensity of moving objects, similar to Kalman filter [38], where the intensity of background models are predicted based on previous information. In addition, these techniques are applied for vehicle tracking in traffic monitoring.

Apart from two motion features mentioned above, an optical flow is also a popular motion based features [23], where the optical flow corresponds to gradient orientations

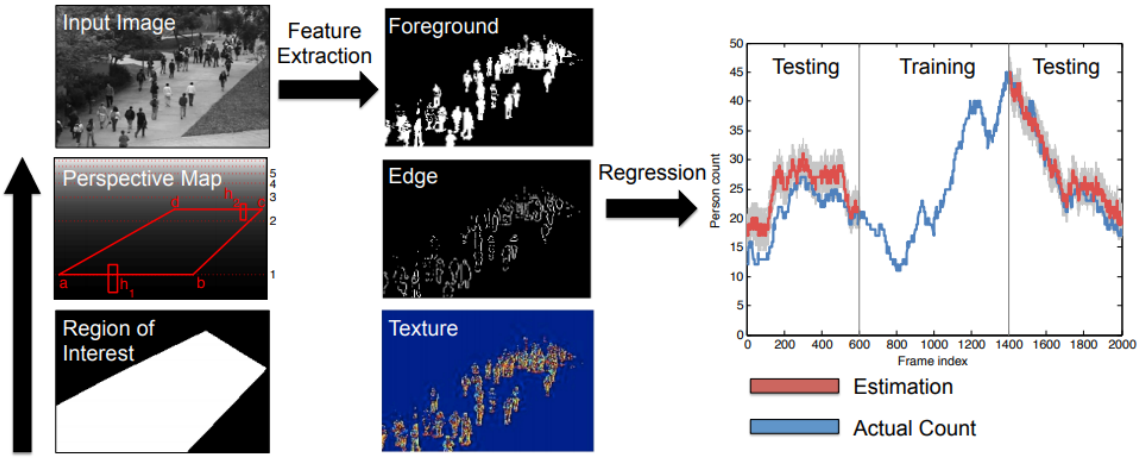



Figure 2.5: A typical pipeline of regression-based approach [50].

and directions of moving objects. The main concept of optical flow is matching two pixel by using temporal and gradient information. Moving objects can be extracted on the high displacement of gradient directions and magnitudes. This feature has a low impact on occlusion. It provides an accurate motion vector which is suitable for a variation of camera motions, lighting conditions, and complex or noisy background. However, its computational complexity is increased because the trajectory is computed from all pixel intensity within images.

## 2.2 Regression-based approach

In recent years, the number of vehicles is increased dramatically which results traffic jam or high vehicle density on the road. This circumstance is called a crowded scene where vehicle regions are extremely overlapped. A detection-based approach suffers from object occlusion and various their scales in high density. To solve this problem, the regression-based approach was designed to handle crowd scenes. Instead of extracting target features separately, the regression-based approach avoids actual segmentation. The regression-based approach relies on holistic features which represents characteristics of object density (e.g. foreground, edge, and texture) as shown in Figure 2.5. The properties of holistic features (e.g. the foreground area and the number of edge) are derived within a perspective map or a region of interest (ROI) as depicted in Figure 2.5. A regression model has a role to establish a direct mapping of property to the number of object in the image.

### 2.2.1 Holistic feature representation

Similar to a detection-based approach, holistic features must be defined for crowd representation or abstraction. Feature representation concerns about the extraction, selection, and transformation of low-level visual properties in an image or video to construct intermediate input to a regression model. The holistic features are categorized as follows:

- **Foreground segment features** are utilized in most previous works for counting object in crowd scenes. These features are obtained by background subtraction in

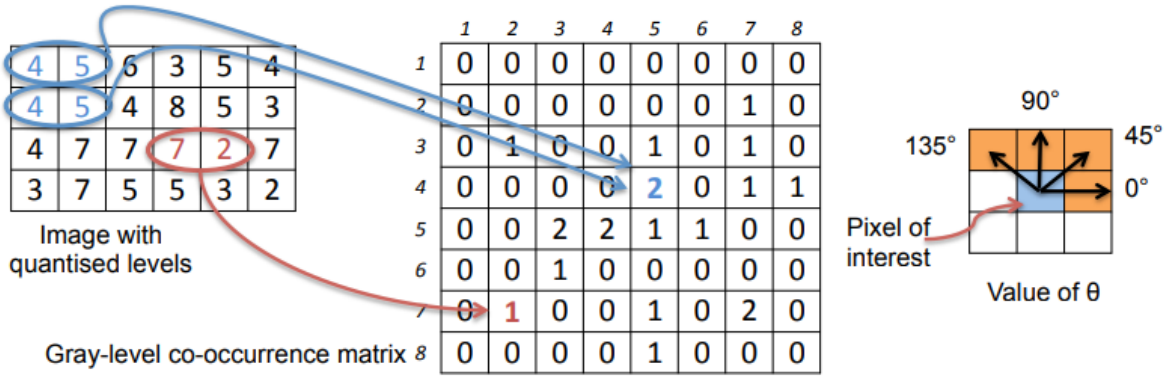



Figure 2.6: The example of Gray-level co-occurrence matrix [50].

motion-based methods (e.g. mixture of Gaussian-based technique [77], mixture of dynamic textures-based method [12], and so on) and semantic segmentation [51]. Various holistic features can be derived from the extracted foreground segment (e.g. area, perimeter, block count and so on). Even though foreground segment features are popular in a regression-based approach, background subtraction is sensitive to lighting conditions.

- **Edge features** carry complementary information of the local and internal patterns [11]. Object density can be described by edge representation, where images with low and high density present coarse and complex edges, respectively. Since edge features are insensitive to illumination, they can solve a problem related to lighting conditions by foreground segment features. Common edge features (e.g. coarse edges, edge orientation, and Minkowski dimension [55]) are usually employed for counting objects.

- **Textured and gradient features** visualize the spatial information of objects in the scenes. Crowded texture and gradient patterns carry strong cues about the number of obejects in a scene. In particular, high-density crowd regions tend to exhibit stronger texture response with distinctive local structure compared to the low-density region. These features can be represented by two approaches, Gray-level co-occurrence matrix (GLCM) [27] or local binary pattern (LBP) [59], as depicted in Figures 2.6 and 2.7 respectively.

### 2.2.2 Traditional regression model

After feature extraction and perspective normalization, the regression model has an important role to formulate the holistic features and predict the number of objects within the regions of interest. A regression model might have a broad class of functional forms as follows:

- **Linear regression** formulates the given feature in form of a linear equation comprising $N$ observations $\{x_n\}$ where $n = 1, 2, 3, ..., N$ corresponding with continuous target values $\{y_n\}$. The goal of this regression is to predict the value of $y$ (the number of objects) for given variable $x$ [8]. The simplest approach is to compute linear

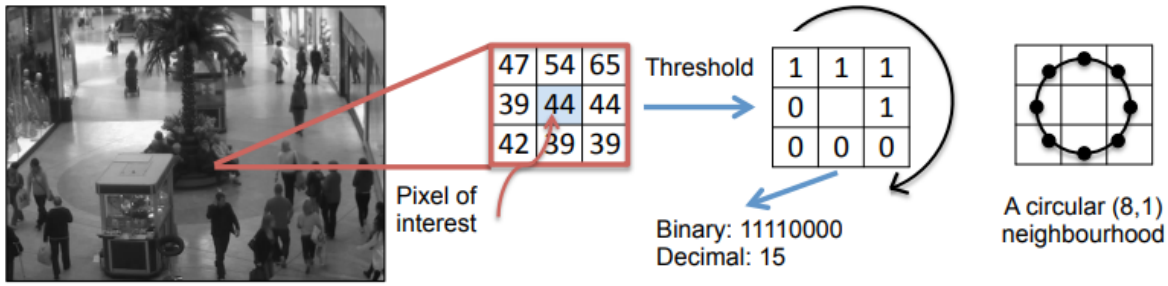



Figure 2.7: The example of local binary pattern operator [59].

regression function $f(x, w)$ that involves a linear combination of the input variables as shown in Eq. (2.1).

$$f(\mathbf{x}, \mathbf{w}) = w_0 x_0 + w_1 x_1 + ... + w_N x_N \tag{2.1}$$

In a sparse scene, linear regression is satisfied for object counting because the mapping between the observations and object count typically presents a linear relationship. Nevertheless, given a more crowded environment with severe inter-object occlusion, it may have to employ a nonlinear regression to adequately capture the nonlinear trend in the feature space. However, linear regression can obtain a large number of unnecessarily observed data $x$ occurring in high-dimensional features, where some of the variables are not effective for prediction. In addition, some of them might be highly co-linear. The unstable estimation of parameters occur [8], leading to very large magnitude in the parameters and therefore a clear danger of severe over-fitting.

In addition, parameters in the linear model is typically obtained by minimizing the sum of square errors in Eq. (2.2), where $\phi(\mathbf{x})$ is a vector of basis function of input vectors ($\mathbf{x}$) and $\mathbf{w}$ is set of weights denoted as $w_0, w_1, w_2, ..., w_N$. Previous works usually deploy Gaussian and sigmoidal basis functions for their works.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{n=0}^{N-1} \{y_n - \mathbf{w}^T \phi(\mathbf{x})\} \tag{2.2}$$

- **Partial least squares regression (PLSR)** [25] addresses the problem of the co-linearity from linear regression. PLSR projects both input $\mathbf{X} = \{x_n\}$ and output $\mathbf{Y} = \{y_n\}$ into a latent space, with a constraint such that the lower-dimensional latent variables describe the covariance between $\mathbf{X}$ and $\mathbf{Y}$ as much as possible. Formally, the PLSR decomposes the input and target variables as described in Eqs. (2.3) and (2.4).

$$\mathbf{X} = \mathbf{T}\mathbf{P}^T + \epsilon_x \tag{2.3}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^T + \epsilon_y \tag{2.4}$$

where $\mathbf{T}$ and $\mathbf{U}$ are score matrices, with the column of $\mathbf{T}$ being the latent variables of $\mathbf{X}$ and $\mathbf{Y}$, respectively $\mathbf{P}$ and $\mathbf{Q}$ are known as loading matrices [25] and $\epsilon$ is the error terms. This decomposition are made to maximize the co-variance of $\mathbf{T}$ and $\mathbf{U}$.

- **Kernel ridge regression** is another solution for relaxing co-linearity problems. A regularization term to error function of linear regression was added shown in Eq. (2.2). A simple regularization is given as $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ as written in Eq. (2.5).

$$E_R(\mathbf{w}) = \frac{1}{2}\sum_{n=0}^{N-1}\{y_n - \mathbf{w}^T\phi(\mathbf{x})\} + \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} \tag{2.5}$$

  where $\lambda$ is a constant value to control the trade-off between the penalty and the fit which is determined by cross-validation. Using this particular choice of regularization term, a error function of ridge regression ($E_R$) is obtained.

  Kernel ridge regression is a non-linear version of ridge regression which is achieved by kernel trick [71]. This technique is used for enlarging dimensional feature space by defining inner product between basis functions shown in Eq. (2.6) where $k(\mathbf{x}, \mathbf{x}^{'})$ is a kernel function which has typical choices of linear, polynomial, or radial basis function (RBF) kernels. The regression function is illustrated in Eq. (2.7), where $\alpha$ are Lagrange multipliers and f$(\mathbf{x}, \alpha)$ is a regression model.

$$k(\mathbf{x}, \mathbf{x}^{'}) = \phi(\mathbf{x})^T\phi(\mathbf{x}^{'}) \tag{2.6}$$

$$f(\mathbf{x}, \alpha) = \sum_{n=0}^{N-1}\alpha_n k(\mathbf{x}, \mathbf{x}_n) \tag{2.7}$$

- **Support vector regression** [91] focuses on reduction of time complexity in Eq. (2.7) which achieves sparseness in $\alpha$ in Eq. (2.7) by applying the concept of support vectors to determine the solution which can result in faster testing speed than Kernel ridge regression that sums over the entire training set. Its regression function can be written in Eq. (2.8).

$$f(\mathbf{x}, \alpha) = \sum_{SVs}(\alpha_n - \alpha_n^*)k(\mathbf{x}, \mathbf{x}_n) + b \tag{2.8}$$

  where $\alpha_n$ and $\alpha_n^*$ are represented as Lagrange multipliers, $k(\mathbf{x}, \mathbf{x}_n)$ denotes the kernel, and $b$ is the offset. The error function for this regression is $\epsilon$-insensitive error function [81].

- **Gaussian processes regression** [69] is one of the most popular nonlinear regression model for counting objects in images with high density. It allows the possibly infinite number of basis function driven by the data complexity, and it models uncertainty in regression problems elegantly, where regression can be written in Eq. (2.9).

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}^{'})) \tag{2.9}$$

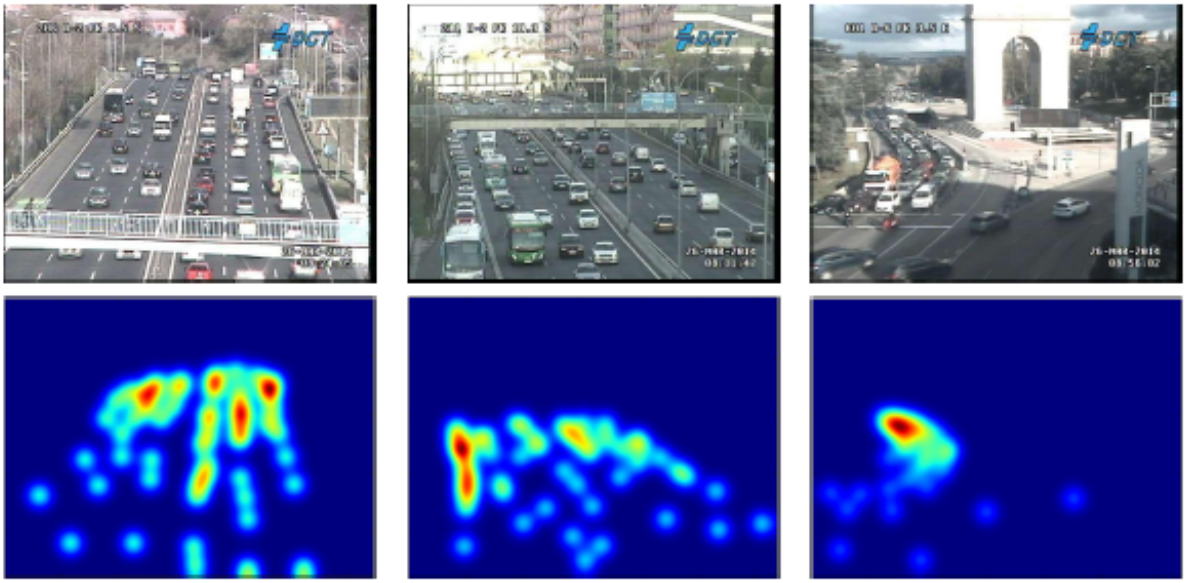


Figure 2.8: The example of input data of regression model based on CNN where RGB images (top) and their density maps (bottom) [64].

where $GP$ is a Gaussian processes specified by mean function ($m(\mathbf{x})$ and co-variance function or kernel ($k(\mathbf{x}, \mathbf{x}^{'})$). The main disadvantage of Gaussian processes regression is its poor tractability for a large number of training sample.

- **Random forest regression** [17] is scale-aware nonlinear regression. A random forest consists of randomly trained regression trees, which achieve better generalization than a single over-trained tree. Each tree in a forest splits a complex nonlinear regression problem into a set of sub problems, which is handled by weak learners such as a linear model. To train a forest, one optimizes an energy over a given training set and associated values of target variable. The regression function was computed by averaging individual posterior distribution as shown in Eq. (2.10).

$$f(\mathbf{x}) = \frac{1}{T} \sum p_t(y|\mathbf{x}) \tag{2.10}$$

where $T$ is the total number of trees in the forest, $p_t(y|\mathbf{x})$ is the posterior of $t^{th}$ tree. The weakness of Random forest regression is that it is poor in extrapolating points beyond the value range of the target variable within the training data.

### 2.2.3 Regression model based on CNN

In recent years, CNN is also usually deployed in regression-based approach with promising counting accuracy performance. Instead of using holistic features, input images are formulated as a non-linear function corresponding to a number of objects called 'density map' as illustrated in Figure 2.8. The density map visualize the level of density where high values are represented as high density. Regression model based on CNN utilizes deep

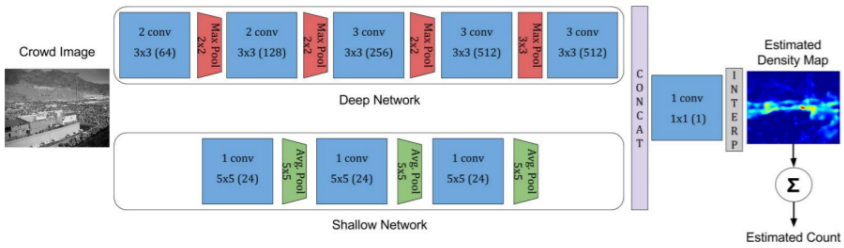



Figure 2.9: Overview of counting method [9] where deep network (top) is used in combination with a shallow network (bottom) to address scale variation across images.

learning techniques for learning non-linear functions from the crowd scene to their corresponding density maps or corresponding counts. Based on the property of the regression model, we classify the approaches into the following categories [73]:

- **Common CNNs** utilize basic CNN layers in their networks. These standard CNN usually be deployed as an initial deep learning approach for crowd counting and density estimation.
- **Scale-aware models** modify object into more sophisticated models that are insensitive to camera viewpoints and scale. This robustness is achieved by different techniques (e.g. multi-column, multi-resolution networks, and so on).
- **Context-aware models** focus on local and global contextual information present in the images for improving counting accuracy.
- **Multi-task frameworks** deploys various techniques beside object counting to achieve low estimation errors (e.g. foreground-background subtraction, velocity estimation, and so on).

Since our goal focused on the evaluation of various condition in traffic videos, scale-aware models are only described in this section which is summarized as follows:

1. Boominathan et al. [9] combine deep and shallow fully convolutional networks to predict the density map for a given crowd image as shown in Figure 2.9. The combination of two networks gives the robustness to non-uniform scaling of crowd and shooting conditions. Patches from the multi-scale image representation are sampled to make the system robust in scale variations.
2. Zhang et al. [95] propose multi-column based architecture (MCNN) as shown in Figure 2.10for solving various density and caemra shooting conditions. The proposed method ensures robustness to a large variation in object scales by constructing a network consisting of three columns corresponding to filters with receptive fields of different sizes (e.g. large, medium, small, and so on). These different columns are designed to provide different object scales present in the images.
3. Kumagai et al. [42] utilize multi-column networks depicted in Figure 2.11 for handling the various scale of a target while aiming of predictor for selecting suitable networks for prediction. The architecture consists of a mixture of expert CNN and

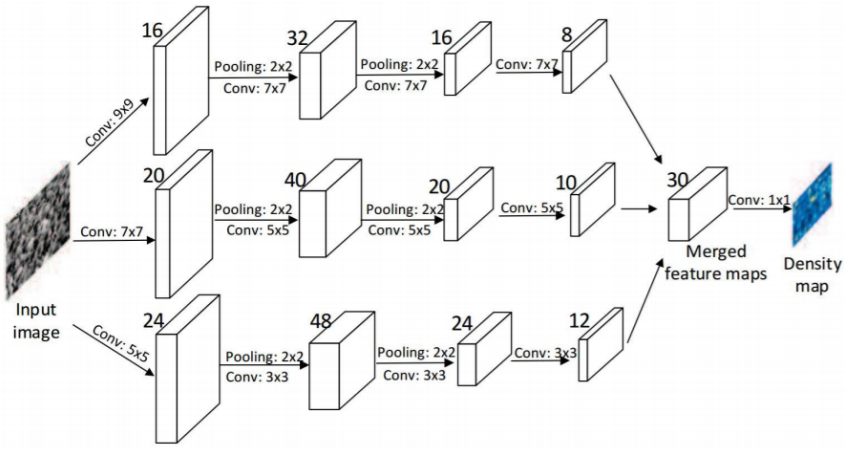



Figure 2.10: Overview of single image crowd counting via multi-column network [95]

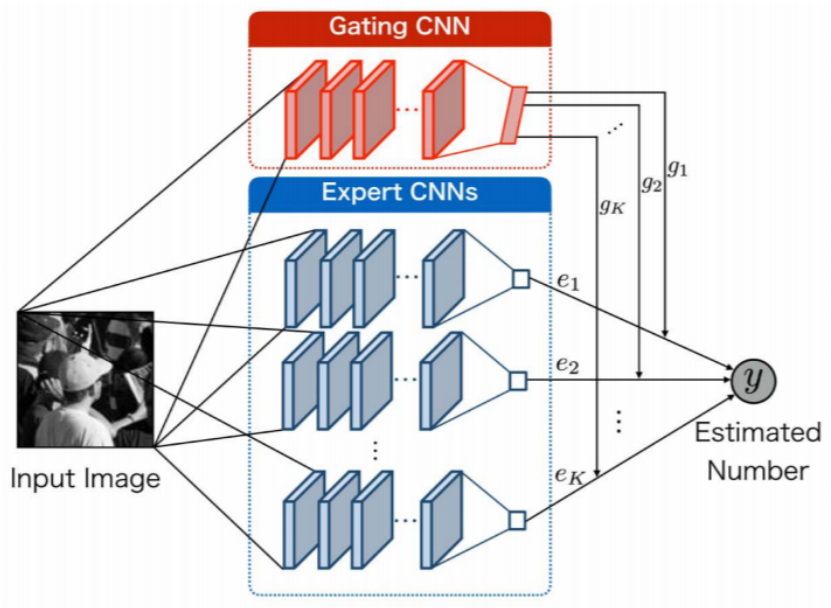



Figure 2.11: Overview of Mixture of CNN for crowd counting [42]

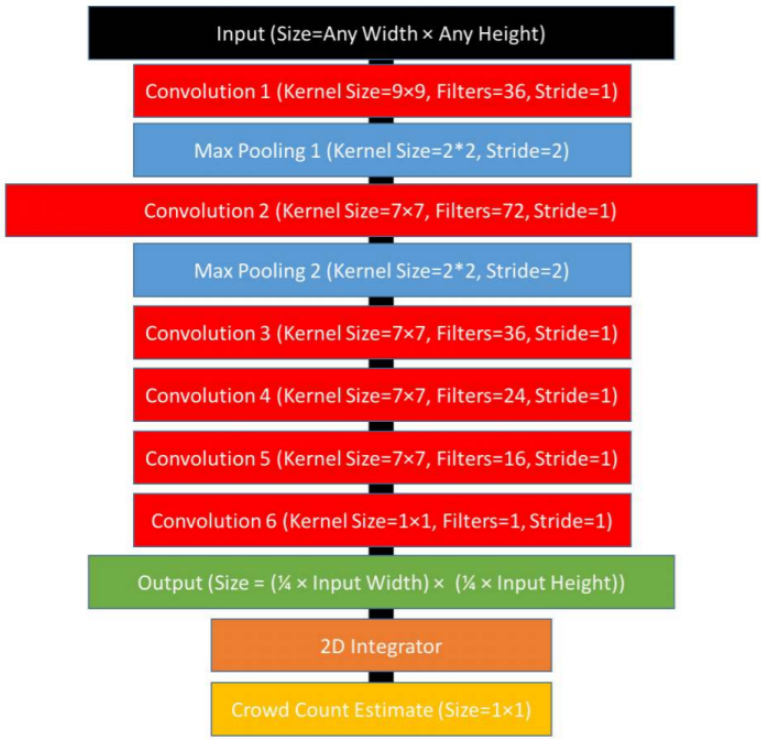



Figure 2.12: Overview of Fully Convolutional Network for crowd counting [56]

a gating CNN that selects the appropriate CNN among the experts according to the appearance of the input image. Gating CNN predicts appropriate probabilities for each of the expert CNN. These probabilities are further calculated as weighting factors to compute the weighted average of the predicted count by all the expert CNNs.

4. Marsden et al. [56] design a deep, single column, fully connected network illustrated in Figure 2.12 utilizing for generating crowd density maps. The greater model capacity improves the FCNs ability to learn the highly abstract, nonlinear relationships present in crowd counting datasets.

5. Sam et al. [70] argue the better performance by training regression models with a particular set of training patches by leveraging variation of crowd density within an image. The switch classifier is trained to select the optimal regression model for a particular input patch where the switch and independent regression models are alternatively trained as shown in Figure 2.13.

6. Onoro and Sastre [61] develop Hydra CNN that learns a multi-scale non-linear regression model. The network consists of 3 heads and a body with each head learning features for a particular scale depicted in Figure 2.14. Each head of the Hydra-CNN is constructed using the CNN model whose outputs are concatenated and fed to the body. The body estimates the density map while the different heads extract image descriptors at different scales.

To solve scale problems, multi-column networks and combination of deep and shallow networks were proposed [9, 42, 61, 70, 95]. Their achievements are limited by number

of networks and receptive sizes depended the scales of objects in the dataset, mostly on human or head counting. In addition, they are often difficult to satisfy the real-time application because of large amount of model parameters in multiple networks. Even though [56] is designed as a single column network, it has ineffective results in a testing or inference stage by performing a multi-scale averaging.

## 2.3 Summary

This chapter describes the related work of object counting which are categorized into detection-based and regression-based approaches. A detection-based approach is usually focused by many studies, where this approach extracts target regions (foreground) from background regions. In the vehicle detection, appearance features are usually extracted from low-level features (e.g. color, symmetry, edge, and so on) to high-level features which are obtained from a deep learning technique (RCNN). In addition, vehicle regions are extracted by using the motion features which are obtained from background subtraction, optical flows, and others. Since the detection method with motion features is applied without prior knowledge, they are insensitive to vehicle types because prior knowledge of appearance is not required. However, vehicles with low and high speed cannot be archived by this technique.

On the other hand, regression-based approaches are designed for solving an occlusion problem which occurs in traffic videos with high vehicle density. Instead of extracting vehicle regions, regression-based approaches establish a direct mapping between these properties and the number of objects in the image, while avoiding actual segmentation of individual or track of features. In recent years, CNN has a main role to design a regression-based approach called "prediction model", where a multi-column network was constructed to handle various vehicle sizes and density. However, these techniques have a large number of parameters and computational cost. Therefore, this study is focused on a single-column network which has less computational costs and number of parameters.

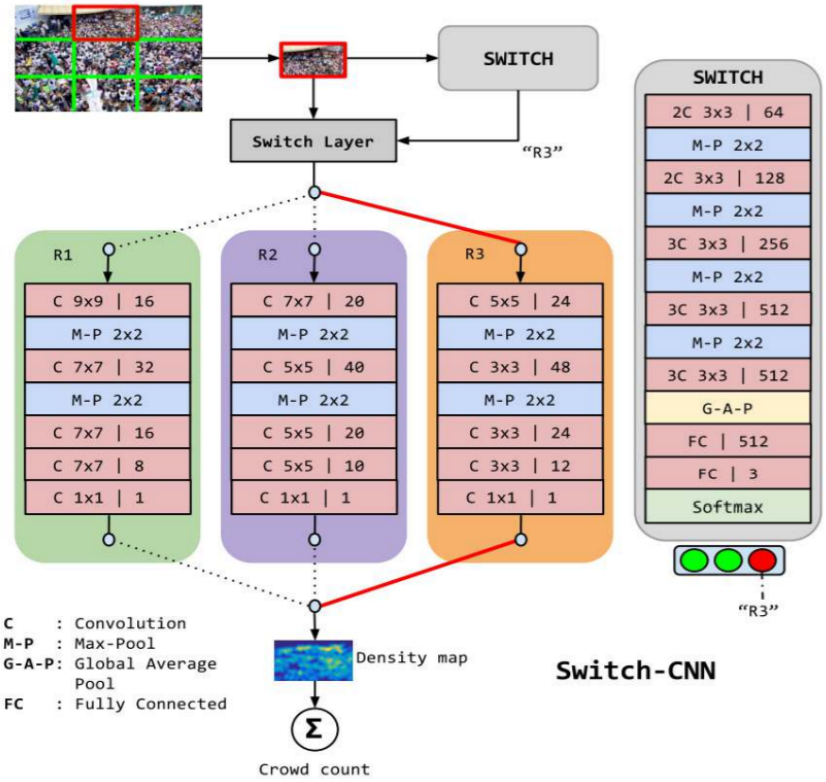



Figure 2.13: Overview of Switching CNN [70]

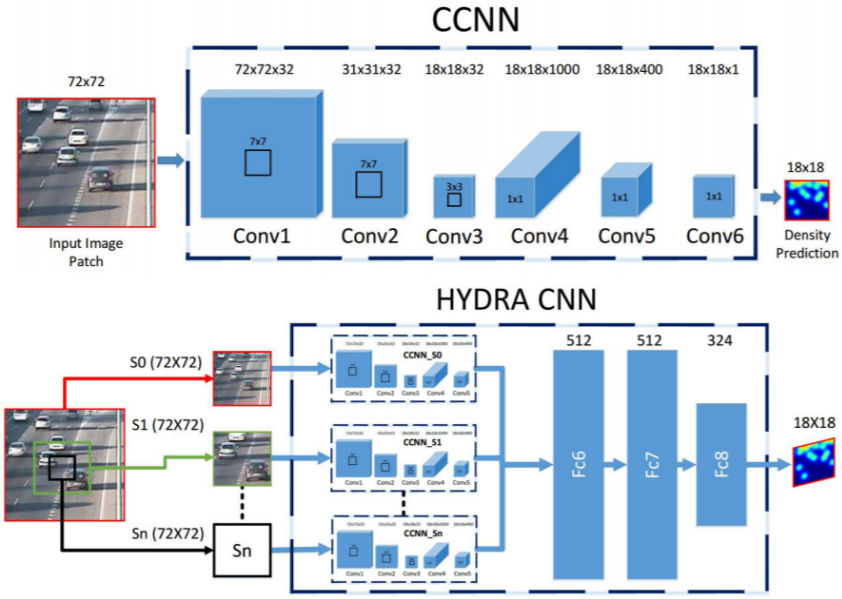



Figure 2.14: Overview of Hydra-CNN [61]

# Chapter 3

# Density Map Pre-processing

To estimate the number of vehicles in given images by applying a regression model based on CNN, the neural configuration should be designed to formulate the output of the regression model. The given outputs are utilized as the ground truth divided into 2 types (estimated count and density map). An estimated count indicates the number of vehicles in given images in form of an integer. On the other hand, a density map visualizes spatial or continuous density information (e.g. the number of the vehicle in one kilometer) and this information is formulated into the number of vehicles. Even though the regression model by estimating vehicle count as output is end-to-end learning, this research is favor on vehicle density map for the following reasons:

- Density map provides more information on partial distribution in specific regions within the given images. It means that the density map is useful in other applications and insensitive to various image conditions containing the same number of vehicles.
- In the learning process by CNN, the model is adapted to the sizes or density of vehicles. Therefore, it is more suitable for arbitrary inputs whose perspective effect varies significantly, and it consequently improves the accuracy of crowd counting.

Since traditional methods for vehicle counting are detection-based approaches, their datasets of traffic videos as shown in Figures 3.1a and 3.1b provide only vehicle locations as ground truth. This ground truth is annotated by bounding boxes (vehicle regions) and dots (vehicle centroids) depicted in Figures 3.1c and 3.1d, respectively. To convert this information into a density map, image processing techniques are deployed where they are described as the following sections:

## 3.1 Traditional density-map computation

As mentioned in the previous section, the ground truth of vehicle detection and counting usually provides us vehicle locations. This information is formulated as delta functions ($\delta$) computed by Eq. (3.1), where $H$ is a centroid matrix, $N$ is the total number of the vehicle in given images, and $(x_i, y_i)$ are vertical and horizontal coordinates of $i^{th}$ vehicles, respectively. To compute density map ($D$), Gaussian kernels ($G$) are convoluted with $H$ as calculated in Eq. (3.2). Gaussian kernels are calculated in Eq. (3.3) where $(S_x, S_y)$ is vertical and horizontal dimensions of input images. As a result, the density maps contain several Gaussian or normal distributions. According to the property of normal

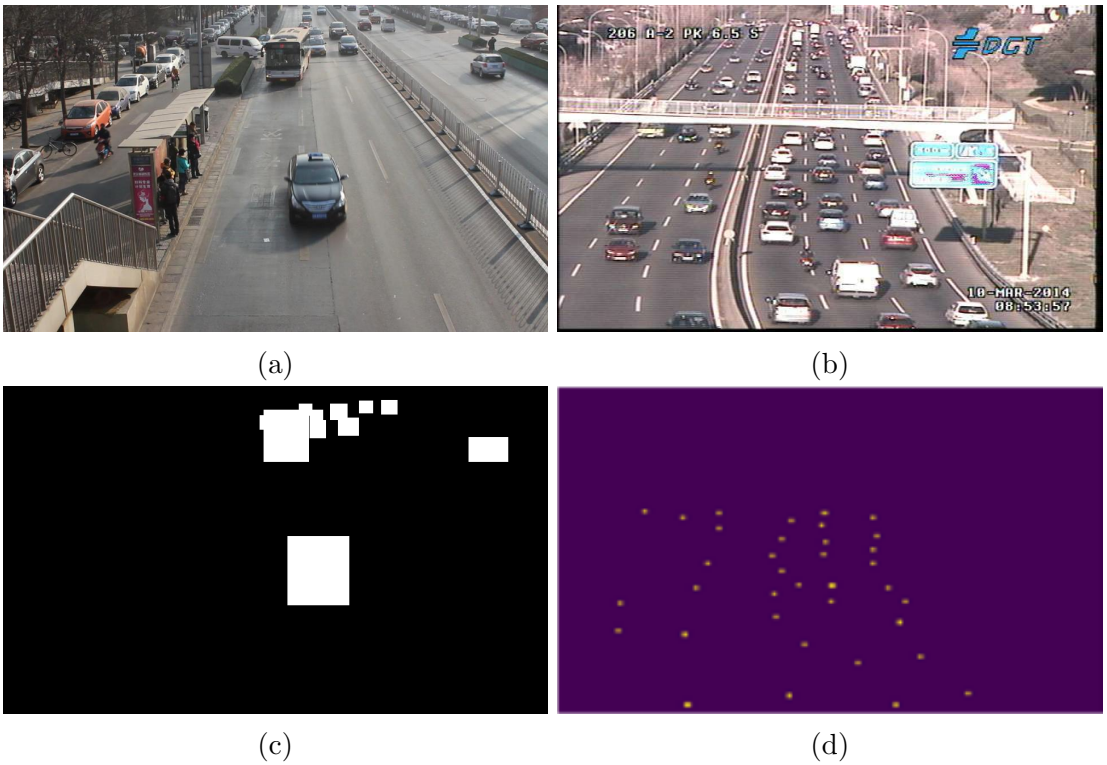



Figure 3.1: Examples of input data for density map computation where (a,b) input images and (c,d) their localization ground truth annotated as bounding boxes and centriods (yellow dots), respectively.

distribution, integration of normal distribution convoluted with delta function is equal to one. Therefore, the number of vehicles or an actual count ($F$) can be calculated by the summation of a density map as shown in Eq. (3.4).

$$H(x,y) = \sum_{i=1}^{N} \delta(x - x_i, y - y_i) \tag{3.1}$$

$$D(x_i, y_i) = H(x,y) * G(x_i, y_i, \sigma) \tag{3.2}$$

$$G(x,y,\sigma) = \sum_{i=1}^{N} \left(\frac{1}{\sigma\sqrt{2\pi}} \sum_{x=1}^{S_x} \sum_{y=1}^{S_y} \{e^{-\frac{(x-x_i)^2(y-y_i)^2}{2\sigma^2}}\}\right) \tag{3.3}$$

$$F = \sum_{x=1}^{S_x} \sum_{y=1}^{S_y} \{D(x,y)\} \tag{3.4}$$

The main factor of the density map relies on spread parameters ($\sigma$). Figure 3.2 depicts density maps calculated by various $\sigma$ and conditions of traffic videos. It shows that larger values of $\sigma$ provide larger distribution areas causing overlapping regions in density maps, especially in traffic images with high vehicle density as shown in $2^{nd}$ and $4^{th}$ columns of Figure 3.2. This configuration reduces the counting accuracy calculated by Eq. (3.4) as shown in Table 3.1. On the other hand, smaller $\sigma$ gives better accuracy. However, the characteristic between regions with high and low density have a similar pattern where

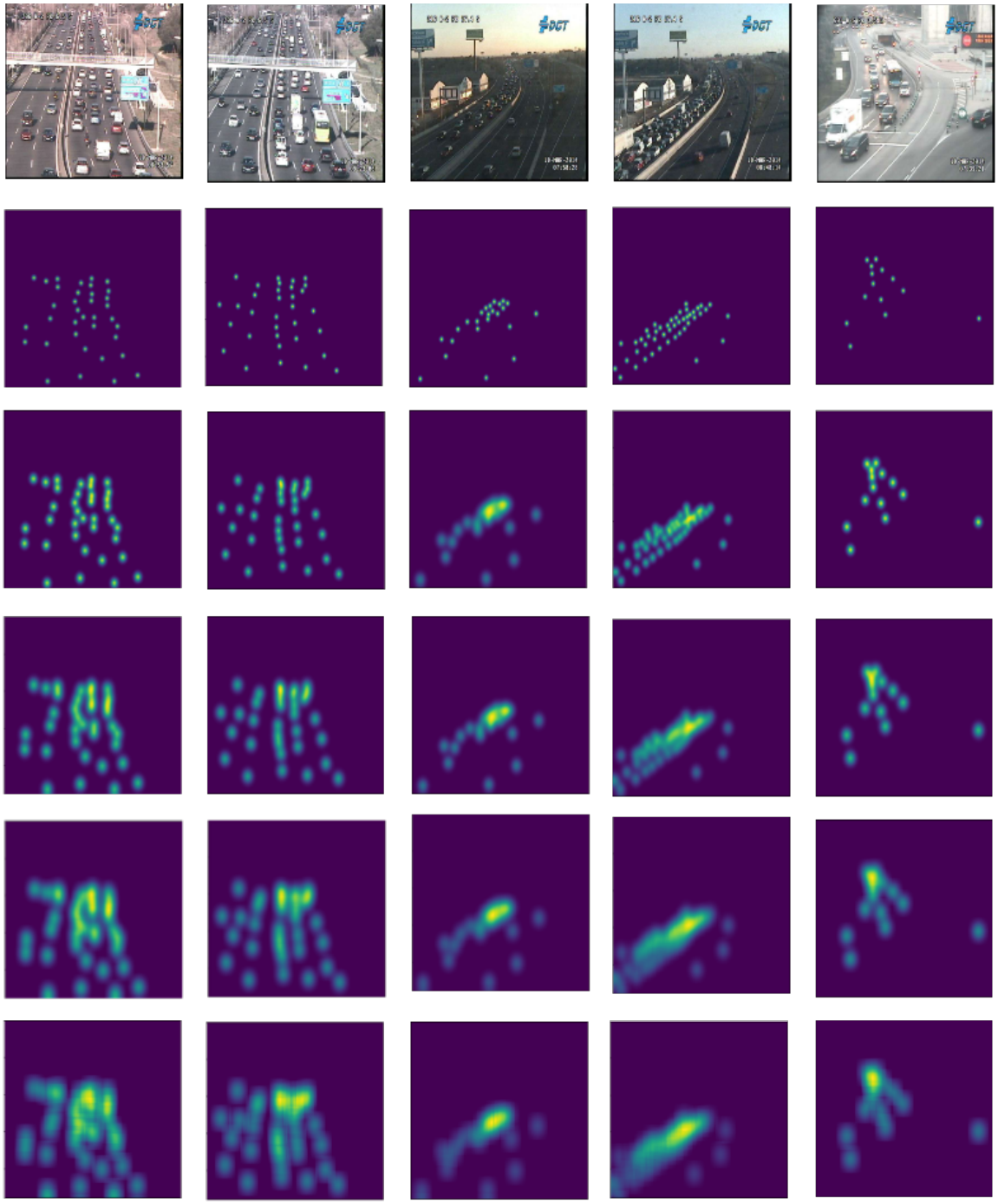


Figure 3.2: Visualization of density maps calculated by using Eq. (3.2) from input images ($1^{st}$ row) with $\sigma = 4$ ($2^{nd}$ row), $\sigma = 8$ ($3^{rd}$ row), $\sigma = 12$ ($4^{th}$ row), $\sigma = 16$ ($5^{th}$ row), and $\sigma = 20$ ($6^{th}$ row).

| Number of vehicles by | $1^{st}$ column | $2^{nd}$ column | $3^{rd}$ column | $4^{th}$ column | $5^{th}$ column |
|---|---|---|---|---|---|
| Ground truth | 36 | 33 | 24 | 46 | 14 |
| $\sigma = 4$ | 36 | 33 | 24 | 46 | 14 |
| $\sigma = 8$ | 36.01 | 32.99 | 24 | 46.03 | 13.99 |
| $\sigma = 12$ | 35.90 | 32.88 | 23.93 | 45.87 | 13.95 |
| $\sigma = 16$ | 34.68 | 31.75 | 23.11 | 44.3 | 13.47 |
| $\sigma = 20$ | 31.80 | 29.11 | 21.19 | 40.62 | 12.35 |

Table 3.1: The summation of the number of vehicle calculated by Eq. (3.2) from traffic images in Figure 3.2

they are hard for training by regression or prediction models. Their empirical results will be discussed in section 3.3.

## 3.2 Density map via geometry-adaptive kernels

In the previous section, the density map was computed by calculating Gaussian kernels. It showed that the density distribution in each region of given images depends on $\sigma$. The performance of counting accuracy also relies on values of $\sigma$. However, the normal distribution computed by traditional methods is overlapped with each other while deploying too large value of $\sigma$, especially in regions with high vehicle density. On the other hand, regions with low vehicle density do not display their continuous density by using low values of $\sigma$. It means that spread parameters should be changed by vehicle density in each region as shown in Figure 3.3, where low and high values of $\sigma$ correspond to high and low vehicle density respectively. Therefore, a modified density map ($D_m$) is written in Eq. (3.5), where $\sigma_i, x_i, y_i$ are a spread parameter, vertical, horizontal coordinates for $i^{th}$ vehicle, respectively.

$$D_m(x_i, y_i) = \sum_{i=1}^{N} \delta(x - x_i, y - y_i) * G(x_i, y_i, \sigma_i) \tag{3.5}$$

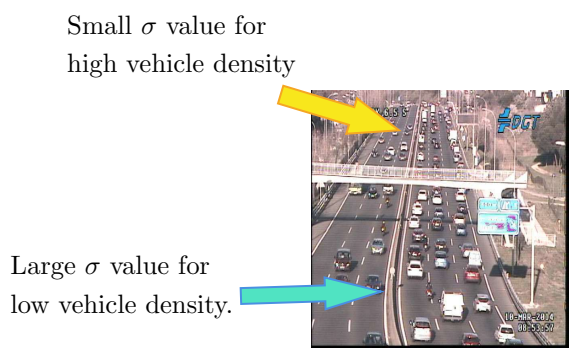



Figure 3.3: The relationship between spread parameters and vehicle density from traffic images with high and low vehicle density[64].

| Number of vehicles by | $1^{st}$ column | $2^{nd}$ column | $3^{rd}$ column | $4^{th}$ column | $5^{th}$ column |
|---|---|---|---|---|---|
| Ground truth | 36 | 33 | 24 | 46 | 14 |
| $\beta = 0.2$ | 36.01 | 32.99 | 23.95 | 46.02 | 13.52 |
| $\beta = 0.4$ | 27.42 | 24.36 | 19.36 | 43.25 | 9.66 |
| $\beta = 0.6$ | 19.28 | 17.02 | 16.55 | 39.37 | 6.98 |
| $\beta = 0.8$ | 13.38 | 11.86 | 13.92 | 34.28 | 4.98 |
| $\beta = 1.0$ | 11.26 | 10.00 | 12.67 | 31.54 | 4.24 |

Table 3.2: The summation of the number of vehicles calculated by Eq. (3.7) from traffic images in Figure 3.4

Since vehicle size is related to vehicle density as shown in Figure 3.3, vehicle size can be taken into account for estimating their spread parameters ($\sigma_i$). However, the vehicle size is not accurately extracted due to the occlusion problem and the variation in a camera viewpoint, and it is also difficult to find the relationship between vehicle size and density map. To solve this problem, the distance among vehicles was utilized as a factor of $\sigma_i$. For each vehicle in given images, a distance to its $k$ nearest neighbors are denoted as $\{ d_1^i, d_2^i, d_3^i, ..., d_k^i \}$. Hence, $\sigma_i$ is evaluated from the average distance between vehicles, which are inversely proportional to vehicle density, as depicted in Eq. (3.6).

$$\sigma_i = \frac{\beta}{k} \sum_{j=1}^{k} d_j^i \tag{3.6}$$

where $\beta$ is a constant factor to control $\sigma_i$. In other words, the centroid matrix is convoluted with density kernels which is adaptive to the local geometry around each data point, referred to as geometry-adaptive kernels. As a result, a modified version of the density map can be used for calculating an actual count ($F_m$) from Eq. (3.7).

$$F_m = \sum_{x=1}^{S_x} \sum_{y=1}^{S_y} \{D_m(x, y)\} \tag{3.7}$$

Similar to the traditional method for computing density maps, $\beta$ is an important factor to indicate the performance of counting accuracy and the value of $\sigma$. Figure 3.4 illustrates the density map implemented by calculating Eq. (3.5) with various $\beta$ and $k = 5$. Regions with high and low vehicle density can be distinguished easily based on their intensities within the density map while utilizing small values of $\beta$. On the other hand, density maps with large values of $\beta$ have high counting errors because of overlapping regions.

## 3.3 Evaluation and comparisons

To select the most suitable density map for prediction models, both techniques for computing density maps with traditional methods and geometry-adaptive kernels, described in previous sections, were evaluated by feeding density maps to a prediction model and comparing their counting accuracy. This evaluation is used to find the best configuration to compute density maps.

In the experiment, traffic images are obtained from the TRANCOS dataset [64] which is a novel benchmark for extremely overlapping vehicle counting in traffic congestion.

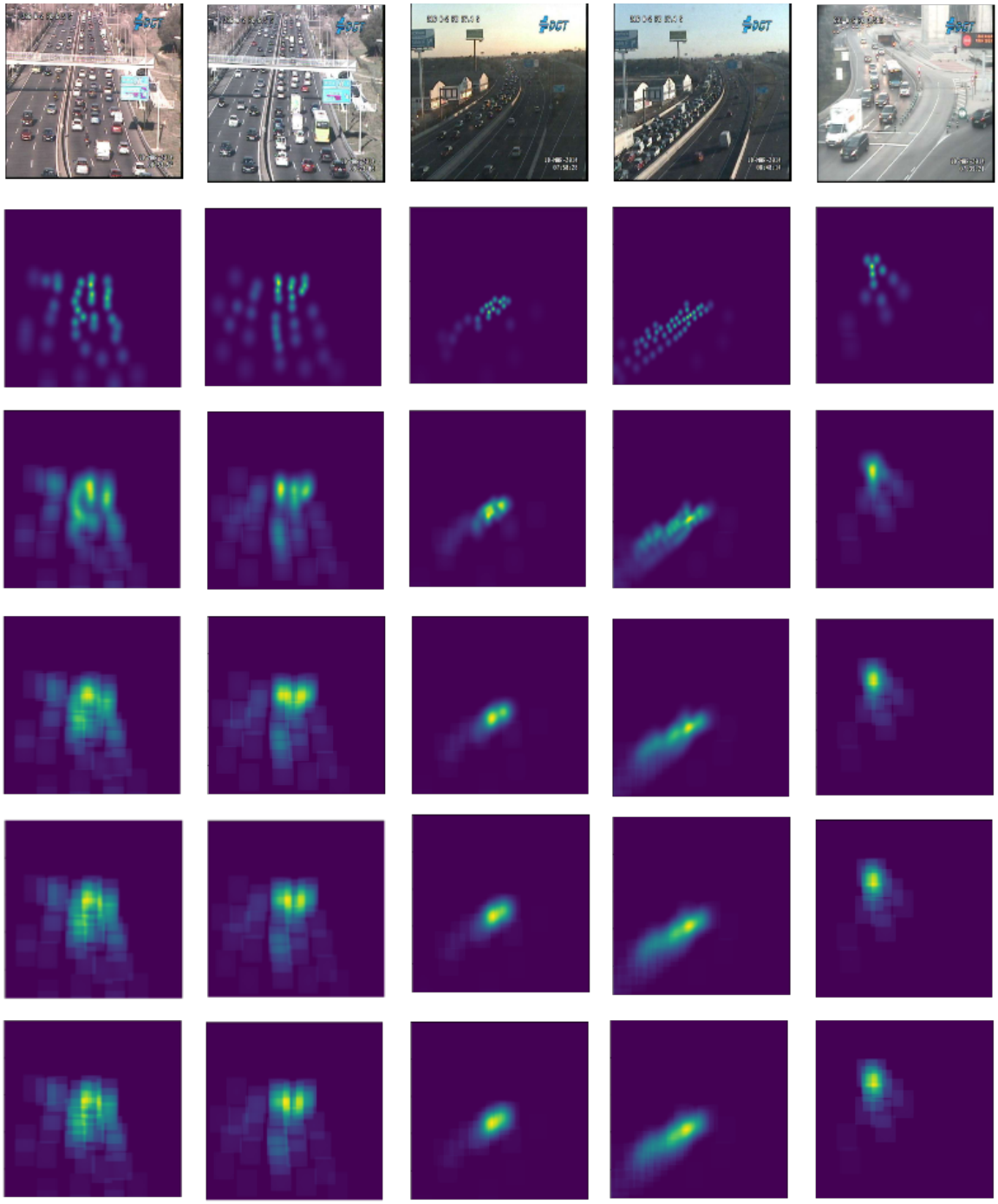


Figure 3.4: Visualization of density maps calculated by using Eq. (3.5) from input images ($1^{st}$ row) with $\beta = 0.2$ ($2^{nd}$ row), $\beta = 0.4$ ($3^{rd}$ row), $\beta = 0.6$ ($4^{th}$ row), $\beta = 0.8$ ($5^{th}$ row), and $\beta = 1.0$ ($6^{th}$ row).

More details are described in chapter 5. The counting accuracy was calculated by mean square error ($MSE$) and mean absolute error ($MAE$) which is the standard evaluation criteria for density map prediction. They were calculated by Eqs. (3.8) and (3.9). $\mathbf{y}$ and $\hat{\mathbf{y}}$ are actual and predicted counts (from prediction model) within images, respectively, calculated from Eqs. (3.4) and (3.7). We deployed MCNN [95] as shown in Figure 2.10, which is the state-of-the-art of crowd counting, for evaluating density maps in a different configuration. All density maps were trained by using the same hyperparameters (the number of epochs = 100 and batch size = 2).

$$MSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{3.8}$$

$$MAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{3.9}$$

### 3.3.1 Counting accuracy evaluation

Figures 3.5 and 3.6 show MSE and MAE from traditional density maps with various spread parameters. As expected, this result shows that $\sigma$ is one of the factors for counting accuracy. Density maps with low ($\sigma < 2$) or high ($\sigma > 2$) values of $\sigma$ gave a high error in both MSE and MAE, where the density map ($\sigma = 2$) is considered as a baseline because it generates the minimum errors. On the other hand, MSE and MAE from the prediction model trained by density map via geometry-adaptive kernels have a similar characteristic, as shown in Figures 3.7 and 3.8, where the density map ($\beta = 0.3$) is considered as a baseline with the same reason. The empirical results show that the density map via geometry-adaptive kernels is more suitable for training in prediction models because of lower minimum error (at their baselines).

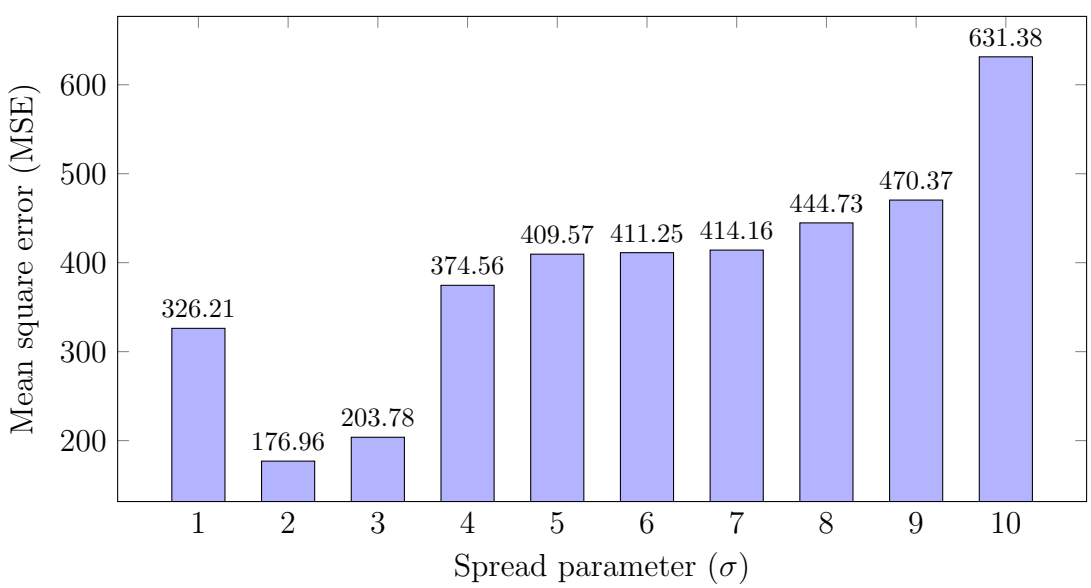



Figure 3.5: Mean square error of traditional density map predicted by MCNN [95].

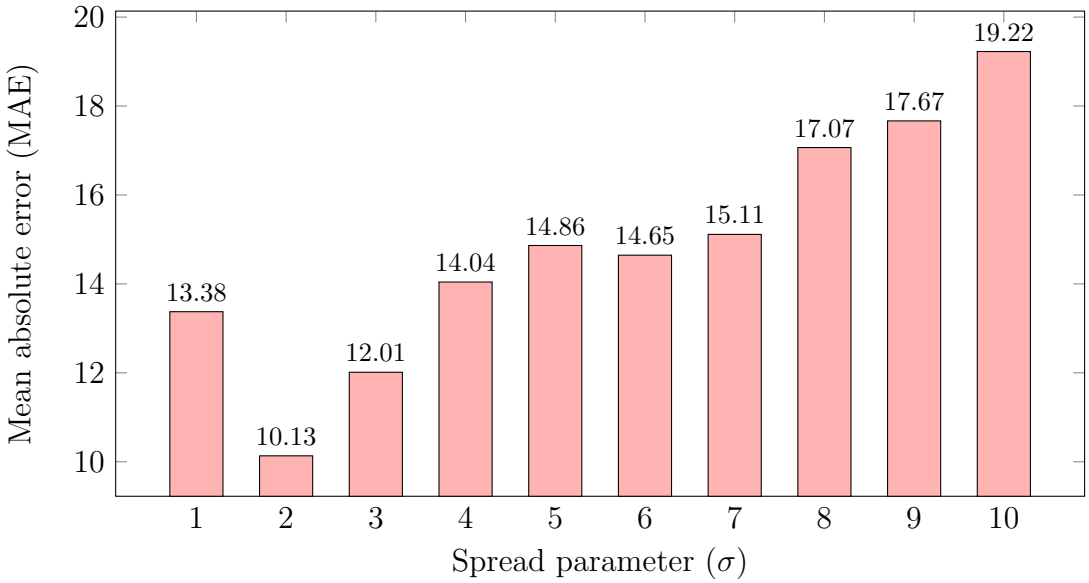



Figure 3.6: Mean absolute error of traditional density map predicted by MCNN [95].

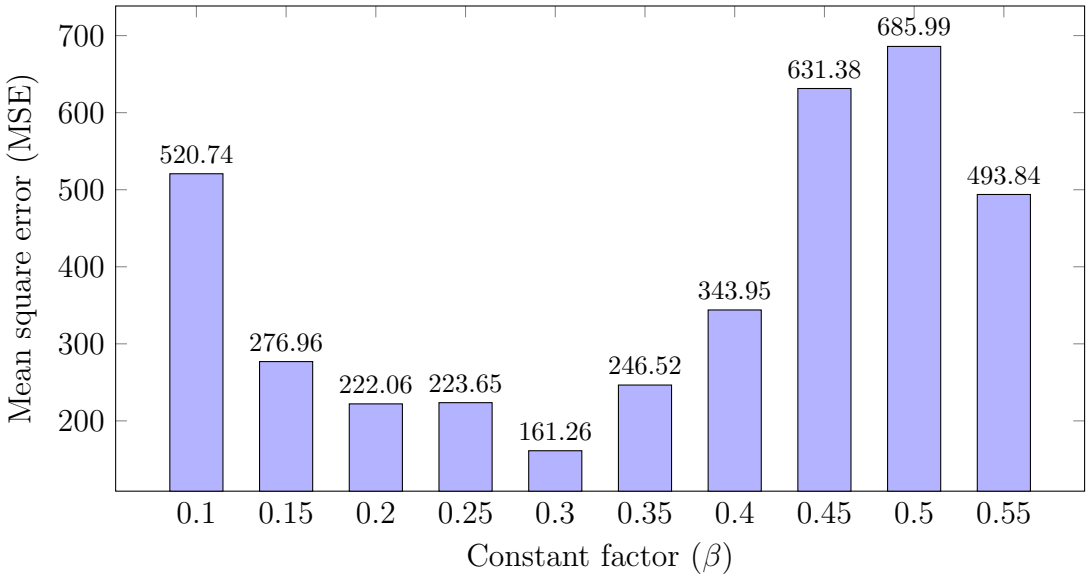



Figure 3.7: Mean square error of density map via geometry-adaptive kernels predicted by MCNN [95].

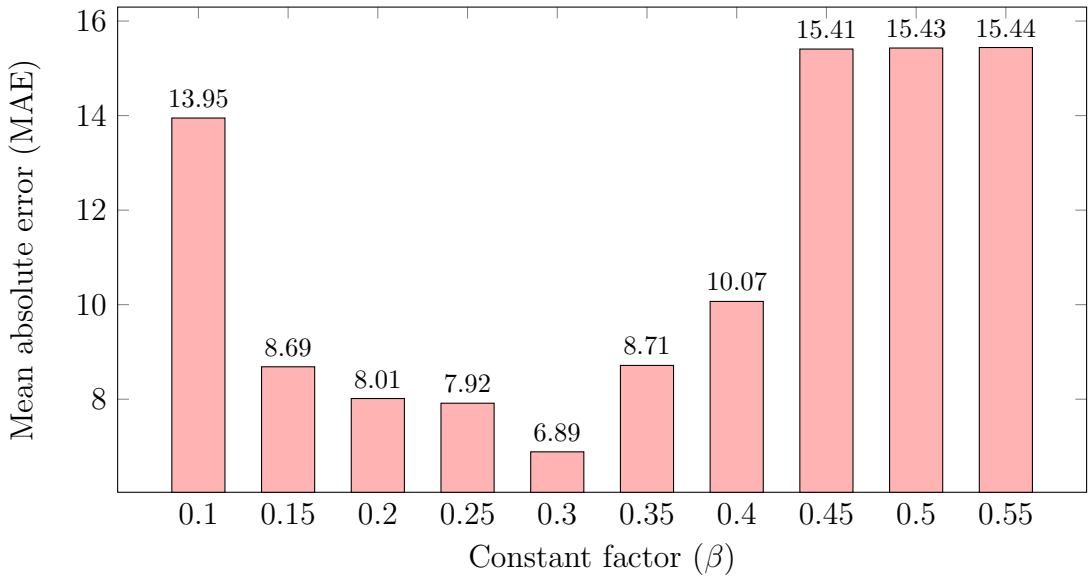



Figure 3.8: Mean absolute error of density map via geometry-adaptive kernels predicted by MCNN [95].

In density maps with high values of $\sigma$ and $\beta$, their errors in MSE and MAE come from the overlapping regions in density maps mentioned in the previous sections. These regions cause counting errors calculated by Eqs. (3.4) and (3.7). On the other hand, predicted density maps generated by applying low values of $\sigma$ and $\beta$ does not distinguish in various vehicle density. More details are discussed in sub-section 3.3.2.

### 3.3.2 Predicted density map assessment

From previous sections, the prediction model was evaluated by the patterns on density maps in different vehicle density or perspective as a factor for density map configuration. Therefore, this section describes the empirical results related to the quality of predicted density maps. As described at the beginning of the chapter 3, actual density maps were employed as ground truth in a loss or cost function while training. It means that a prediction model generates predicted density maps that have the same patterns as actual density maps to reduce training errors.

To evaluate the empirical results, a traffic image containing both low and high vehicle density, as shown inside blue and red circles in Figure 3.9, respectively, is employed by the prediction model. Figure 3.10 depicts traditional density maps (in the first row) and predicted density maps (in the second row) of Figure 3.9. The empirical results show that density maps with $\sigma$ less than baseline ($\sigma = 2$) can not capture the patterns in low vehicle density as shown in Figure 3.10d compared to its actual density map (Figure 3.10a). On the other hand, two other configurations, Figures 3.10e and 3.10f can not extract pattern in high vehicle density compared to their actual density maps (Figures 3.10b and 3.10c). Therefore, counting error in traditional density maps is reduced from losing Gaussian distribution in images, especially in high vehicle density.

For another density map, spread parameters which are controlled by $\beta$ improves quality of predicted maps as shown in Figure 3.11d to 3.11f, especially in baseline ($\beta = 0.2$). However, density maps with $\beta = 0.1$ have the pattern in low vehicle density similar

Figure 3.9: An example of traffic image [64] containing low and high vehicle density within blue and red circles, respectively.

to high vehicle density as shown in Figure 3.11a where their Gaussian distributions are disappeared in some regions. On the other hand, more overlapping regions appear in density maps with a higher value of $\beta$ resulting in incomplete prediction. Their results can be improved by training with more epochs.

## 3.4 Summary

This chapter describes the pre-processing step for a regression-based approach. Since most of the dataset is related to detection-based approaches, their ground truths are typically annotated as bounding boxes (vehicle regions) and dots (vehicle centroids). The actual density map typically are convoluted with Gaussian kernels controlled by spread parameters ($\sigma$) to formulate actual density maps. However, this technique is ineffective for classifying the different information between regions with high and low vehicle density. Therefore, we propose density maps via geometry-adaptive kernels where each location is convoluted with different $\sigma$. The spread parameters of the vehicle ($\sigma$) are controlled by average distances to its nearest neighbors. Both methods are evaluated by feeding them into the prediction model. The empirical result shows that proposed density maps have fewer counting errors and it was applied in this study.

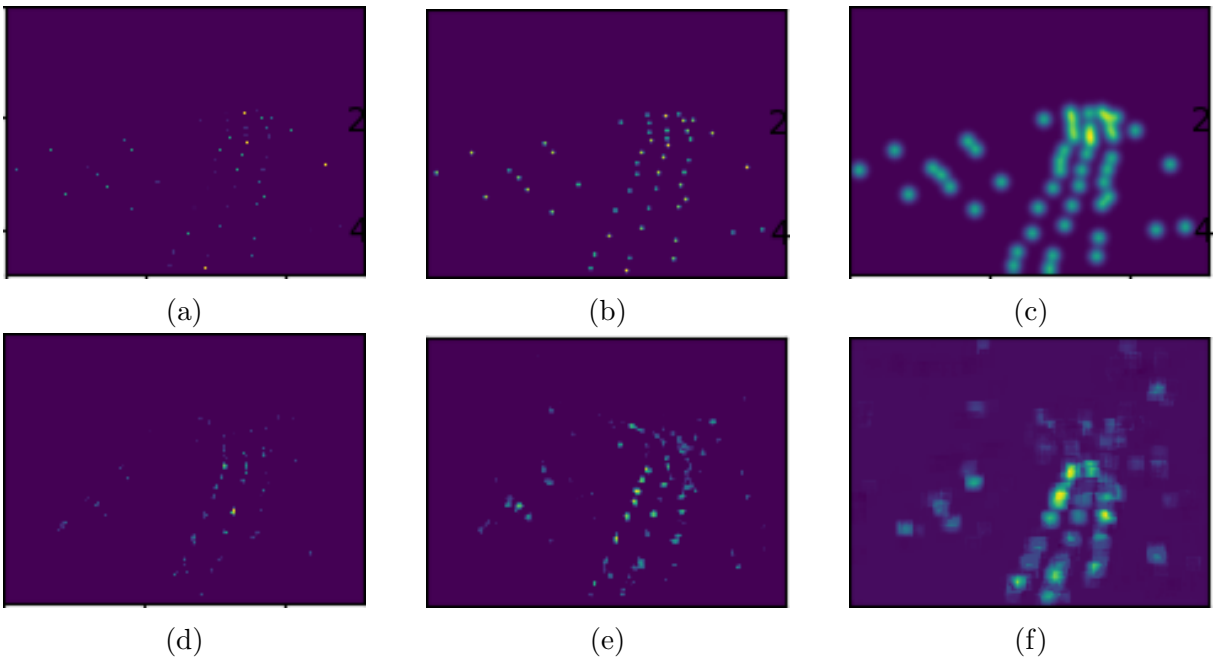



Figure 3.10: Traditional density maps computed from Figure 3.9 with (a) $\sigma = 1$, (b) $\sigma = 2$ (baseline), and (c) $\sigma = 9$, where (d) - (f) are their predicted density maps, respectively.

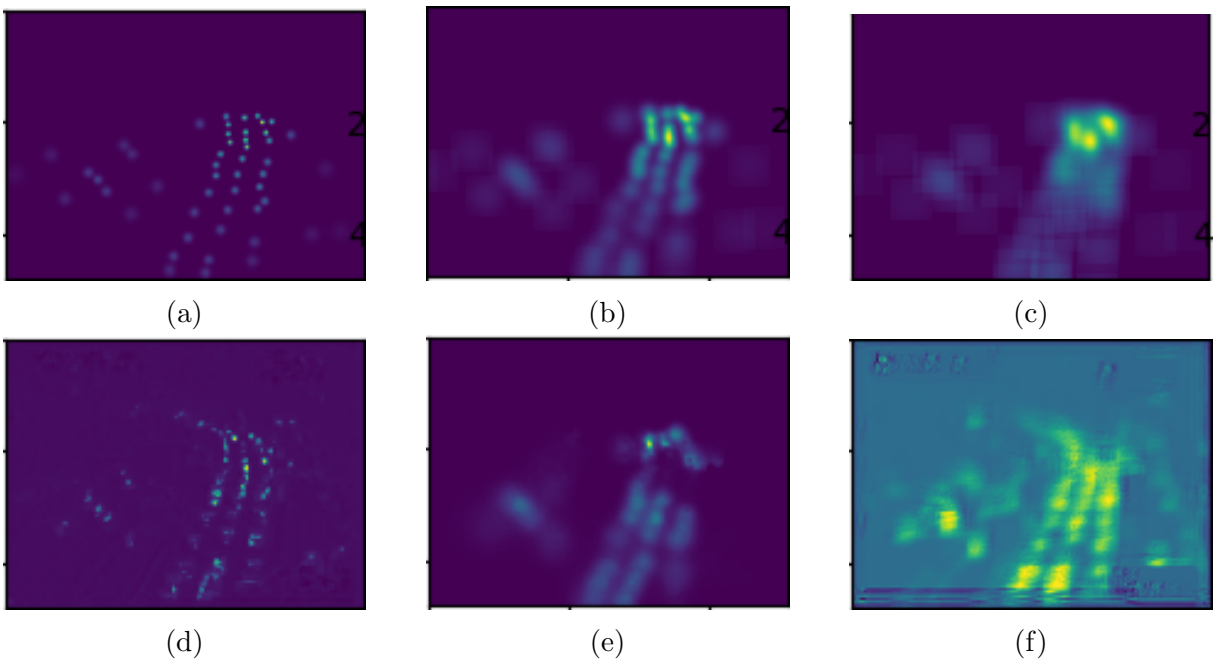



Figure 3.11: Density maps via geometry-adaptive kernels computed from Figure 3.9 with (a) $\beta = 0.1$, (b) $\beta = 0.3$ (baseline), and (c) $\beta = 0.5$, where (d) - (f) are their predicted density maps, respectively.

# Chapter 4

# Vehicle-density Estimation Framework

The prediction model is considered an important part of a regression-based approach because it is end-to-end learning which extracts holistic features and estimates the number of vehicles in images. As mentioned in section 1.3, this study aims to examine the single-column network which has a less computational cost and the number of parameters than the multi-column network. Besides, skip connections and dilated convolutions were also employed to improve counting accuracy and preserve semantic information in a predicted density map, respectively. This chapter describes the prediction model for vehicle-density estimation in the following sections:

## 4.1 Backbone network architecture

First of all, the structure of a prediction model must be indicated (e.g. the number of layers, kernel sizes, and so on). The state-of-the-art prediction model in crowd counting, MCNN [95], was analyzed. Their network architecture is an original concept to apply multiple CNNs to handle objects with different sizes located in different density, a multi-column network. There are three CNNs consisting of three networks to handle objects with small, medium, and large sizes as shown in Figures 4.1a, 4.1b, and 4.1c, respectively. Each network has different kernel sizes to extract an object's features with different sizes. The final results combined from three networks give promising results.

To design the proposed single-column network, the backbone network architecture was selected from one of three networks that have the lowest errors. The most dominated networks are utilized along with skip connections to replace the multi-column network with the single-column network. The dominated network was selected by calculating MSE and MAE in Eqs. (3.8) and (3.9) shown in Table 4.1, where S-net, M-net, and

| MCNN networks | S-net | M-net | L-net |
|---|---|---|---|
| MSE | 185.88 | 146.83 | 148.25 |
| MAE | 11.94 | 10.27 | 10.36 |

Table 4.1: MSE and MAE of MCNN in S-net, M-net, and L-net shown in Figures 4.1a, 4.1b, and 4.1c, respectively.

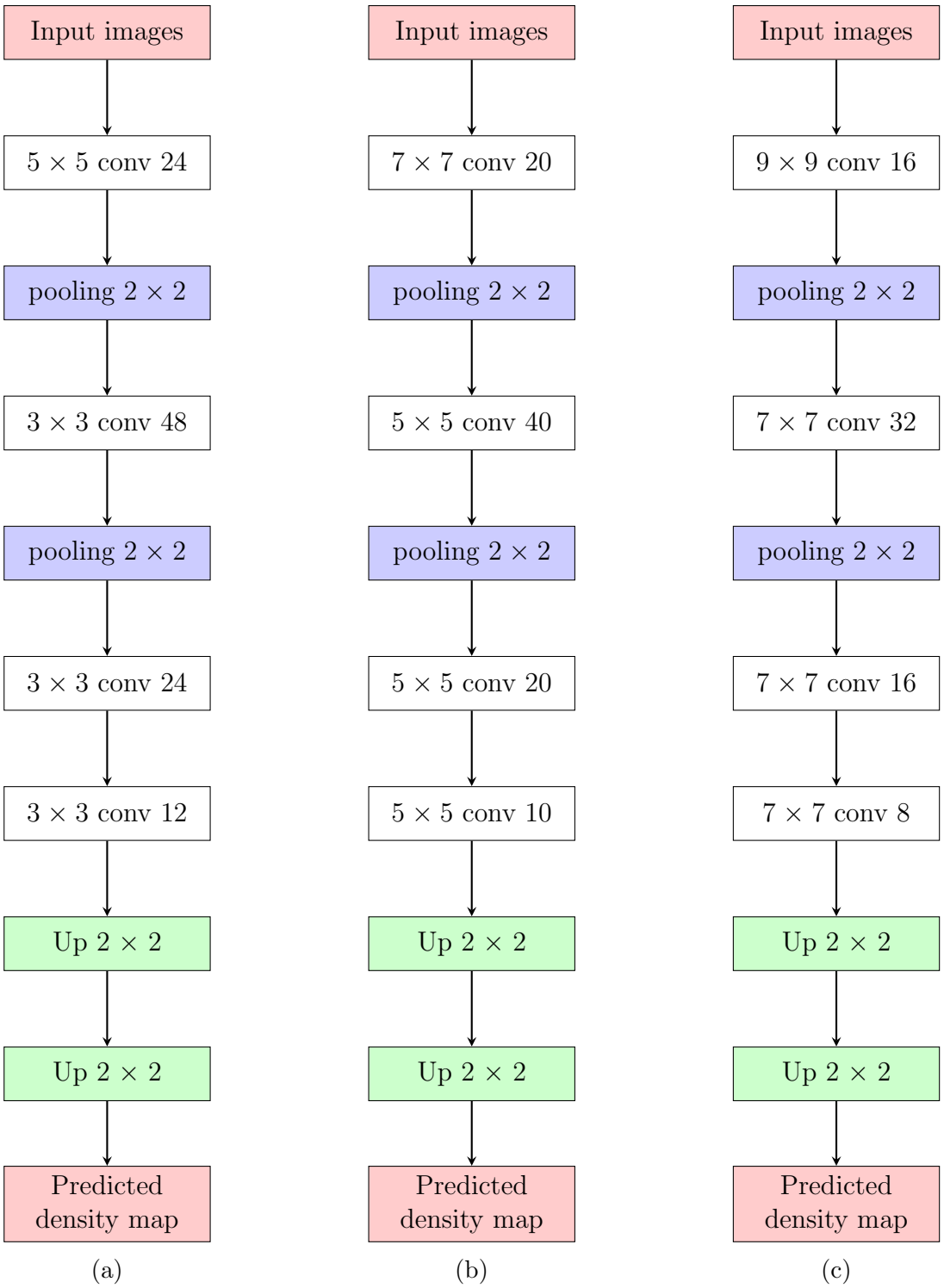



Figure 4.1: CNNs for crowd counting from MCNN [95] which handle objects with (a) small, (b) medium, and (c) large scales.

L-net are corresponded to networks in Figures 4.1a, 4.1b, and 4.1c, respectively. It shows that M-net has the most optimal results. Therefore, M-net was selected in our network. Since one kernel will be computed in a thread of GPU and 8 threads are run in parallel, several kernels should be divisible into 8 to optimize the memory consumption. If the number of kernels is 7, one thread will be a wasted resource. On the other hand, another computation period must be provided if the number of the kernel is more than 8. Besides, Up-sampling layers, as shown in green boxes of Figure 4.1, distort predicted density maps where counting accuracy might be reduced. An example of the distortion is depicted in Figure 4.2, where Figures 4.1a and 4.1b are an input image and its actual density map. it shows that a predicted density map computed by M-net (Figure 4.2c) has quite an agitated pattern compared to its actual density map as selected as a backbone network in the proposed method.

To solve distortion problems, convolutional (Conv) layers are inserted as decoders, as shown in Figure 4.3, to recover the information lost by pooling layers. Figure 4.2d depicts a predicted density map computed by Figure 4.3a. It shows that its pattern is more similar to an actual density map (Figure 4.2b), and the amount of noise is reduced. The proposed backbone network architectures are designed into three versions with different number of kernels, where they are represented as Model A-1 (Figure 4.3a), Model A-2 (Figure 4.3b), and Model A-3 (Figure 4.3c). The counting accuracy is affected by the number of kernels, where their experimental results are discussed in chapter 5.

In the next step, the receptive field and the kernel size will be analyzed. They found that the Conv layer can be replaced with another layer that has the same receptive field size [84]. The receptive field sizes are computed in Eq. (4.1), where $r_n$ is the receptive field size to the $n^{th}$ layer, $k_n$ is the kernel size of $n^{th}$ Conv layer, and $j_n$ is the distance between two adjacent features of $n^{th}$ layer which is calculated by Eq. (4.2) with convolution stride in $n^{th}$ layer ($s_n$).

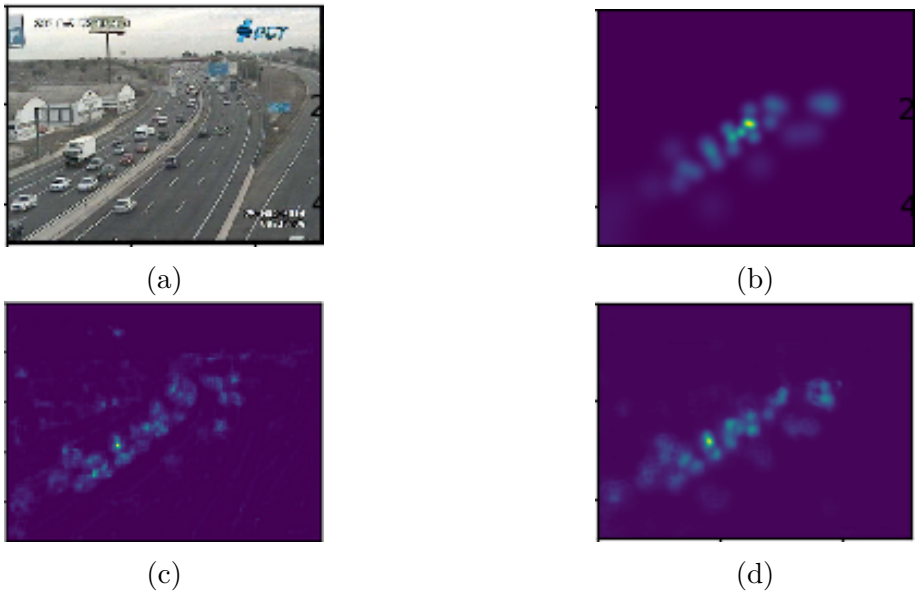
$$r_{n+1} = r_n + (k_{n+1} - 1) \times j_n \tag{4.1}$$

$$j_{n+1} = j_n \times s_{n+1} \tag{4.2}$$

As shown in the equations above, the receptive field relies on the kernel size and the number of Conv layers. It means that backbone network architectures are reconstructed by adding more layers and changing kernel size, which results in the same receptive fields. By employing Eq. (4.1), $7 \times 7$, and $5 \times 5$ Conv layers are converted into stacks of three and two $3 \times 3$ Conv layers, respectively. Figure 4.4 shows an example of equivalent convolution between $5 \times 5$ convolutions and stacked $3 \times 3$ convolutions generating the same output sizes. By using stacks of $3 \times 3$ Conv layers, its advantages are summarized as follows:

- The number of model parameters was reduced due to the smaller sizes of kernels. For example, the number of parameters in $7 \times 7$ Conv layers is 49. On the other hand, only 27 parameters are trained from stacks of $3 \times 3$ Conv layers.
- Even though fewer parameters are provided by stacks of $3 \times 3$ Conv layers, non-linear neuron parameters are calculated where stacks of convolution layers have a higher-level feature map than only one Conv layer.

Figure 4.5 shows modified versions of Models A-1, A-2, and A-3, which are denoted as Models B-1 (Figure 4.5a), B-2 (Figure 4.5b), and B-3 (Figure 4.5c), respectively. These



(a) (b) (c) (d)

Figure 4.2: The experimental result on distortion of predicted density maps where (a) input image, (b) actual, (c), and (d) predicted density maps computed from M-net and Model A-1, respectively.

models were designed by replacing $5 \times 5$ and $7 \times 7$ Conv layers with stacks of $3 \times 3$ Conv layers mentioned above, where a promising performance is expected.

## 4.2 Network architecture with skip connections

To overcome the multi-column network, the feature map in different layers of the single-column network are extracted for handling vehicles located in various vehicle density. Feature maps are categorized into low-level and high-level feature maps. Low-level feature maps represent the simple texture or structure of objects (e.g. lines, corners, gradient, and so on). On the other hand, complicated textures (e.g. faces, whole bodies, and so on) are represented by high-level features. Figure 4.6 illustrates the typical structure of CNN from low-level to high-level features corresponded to shallow and deep layers, respectively. In this study, deep layers are utilized to extract features of large vehicles located in low vehicle density. Shallow layers are more accurate and robust in counting extremely small vehicles with high vehicle density.

To extract a feature map from shallower or deeper layers, skip connections are deployed in the proposed method. It is well known that prediction models using CNNs require a large number of layers to increase counting and classification performance. Instead of designing deep networks, the skip connection was proposed by reusing low-level feature maps, where its advantages are summarized as follows:

- Information degradation is reduced during the training process similar to the residual block identity mapping.
- The low information for prediction is recovered by integrating a low-level feature map from a shallower layer.

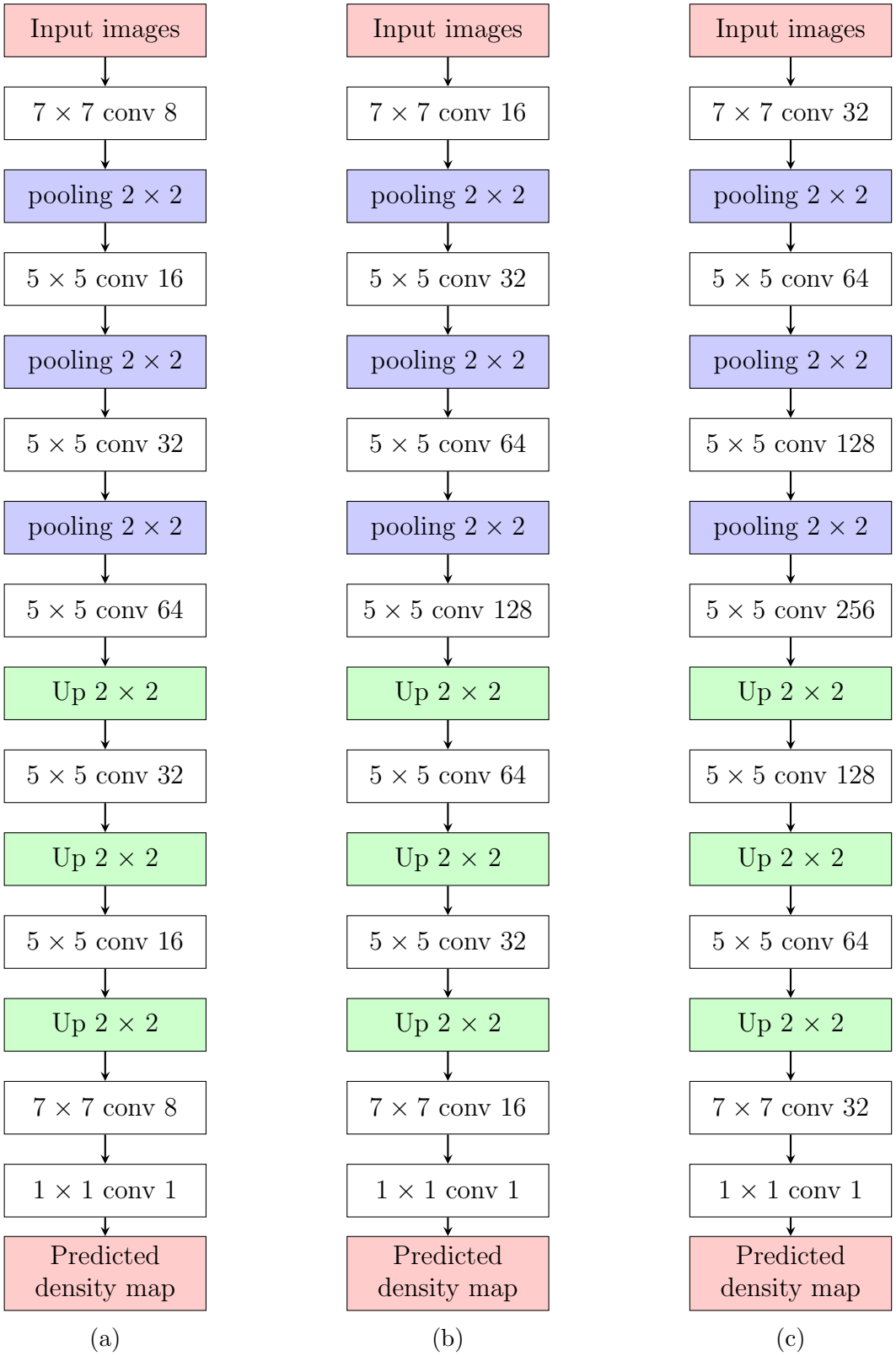



Figure 4.3: The proposed backbone network for solving distortion problems consisting of 3 versions, (a) Model A-1, (b) Model A-2, and (c) Model A-3.

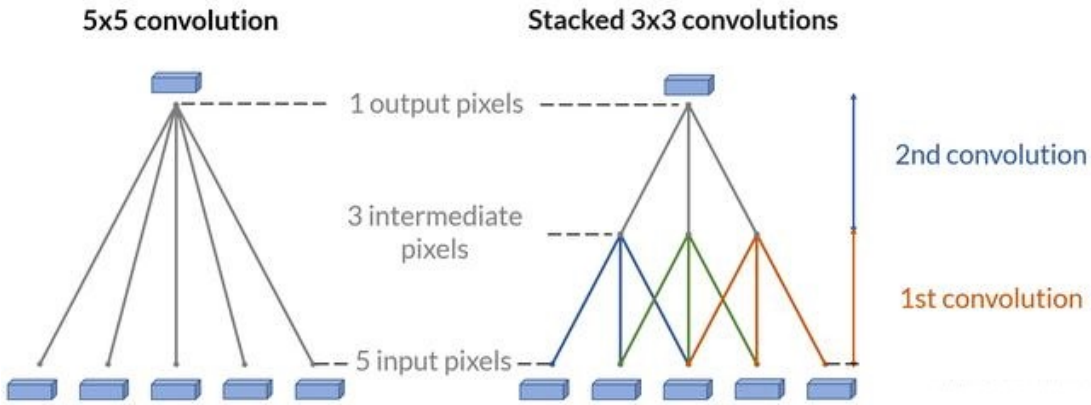



Figure 4.4: $5 \times 5$ convolutions vs the equivalent stacked $3 \times 3$ convolutions [19]

CNNs with skip connections usually extract low-level features from the shallower layer to deeper layers, for reconstructing high-level features. It is called as forward connections. However, CNNs with forward connections may not aware of information from deep layers and it is very hard for training. If high-level features maps are fed by shallow layers, more informative features are extracted in the earlier stage using backward connections, extracting high-level features from deeper layers to shallower layers. Their information is described as follows:

### 4.2.1 Forward skip connection

As described in the previous section, a forward skip connection extracts feature maps from shallower layers to increase prediction and counting accuracy. Feature maps from shallower and deeper layers are combined by concatenation as shown in Eq. (4.3), where $x_L$ is the input of $L^{th}$ Conv layer resulting $y_L$, $n$ is a scalar constant value, and $F(x_L)$ is a Conv layer function that consists of multiple convolutions and non-linear activation functions (ReLU).

$$y_L = F(x_L) + W_L x_{L-n} \tag{4.3}$$

To concatenate two feature maps, their sizes must be the same. It denotes that $\{H_L, W_L, D_L\}$ represents as height, width and depth. Then, $W_L$ is added into Eq. (4.3) to adjust $H$ and $W$ of the feature map, where $W_L$ is a convolutional transformation of the $L^{th}$ layer. Since the research goal of this study is to minimize time complexity and the number of model parameters, $W_L$ should be ignored. The proposed prediction model is designed from U-net [66], where feature maps with the same sizes are concatenated to each other. Therefore, Eq. (4.3) are written as Eq. (4.4) without $W_L$, where $N_L$ is the number of convolution layers in prediction models.

$$y_L = F(x_L) + x_{N_L - L} \tag{4.4}$$

Figure 4.8 depicts the prediction model with a forward connection, where $f$ is a factor to control the number of the kernel in each Conv layer. It shows that the feature maps from encoded Conv layers (Conv1, Conv2, and Conv3) are concatenated with input feature maps of decoded Conv layers (Conv5, Conv6, and Conv7), respectively.

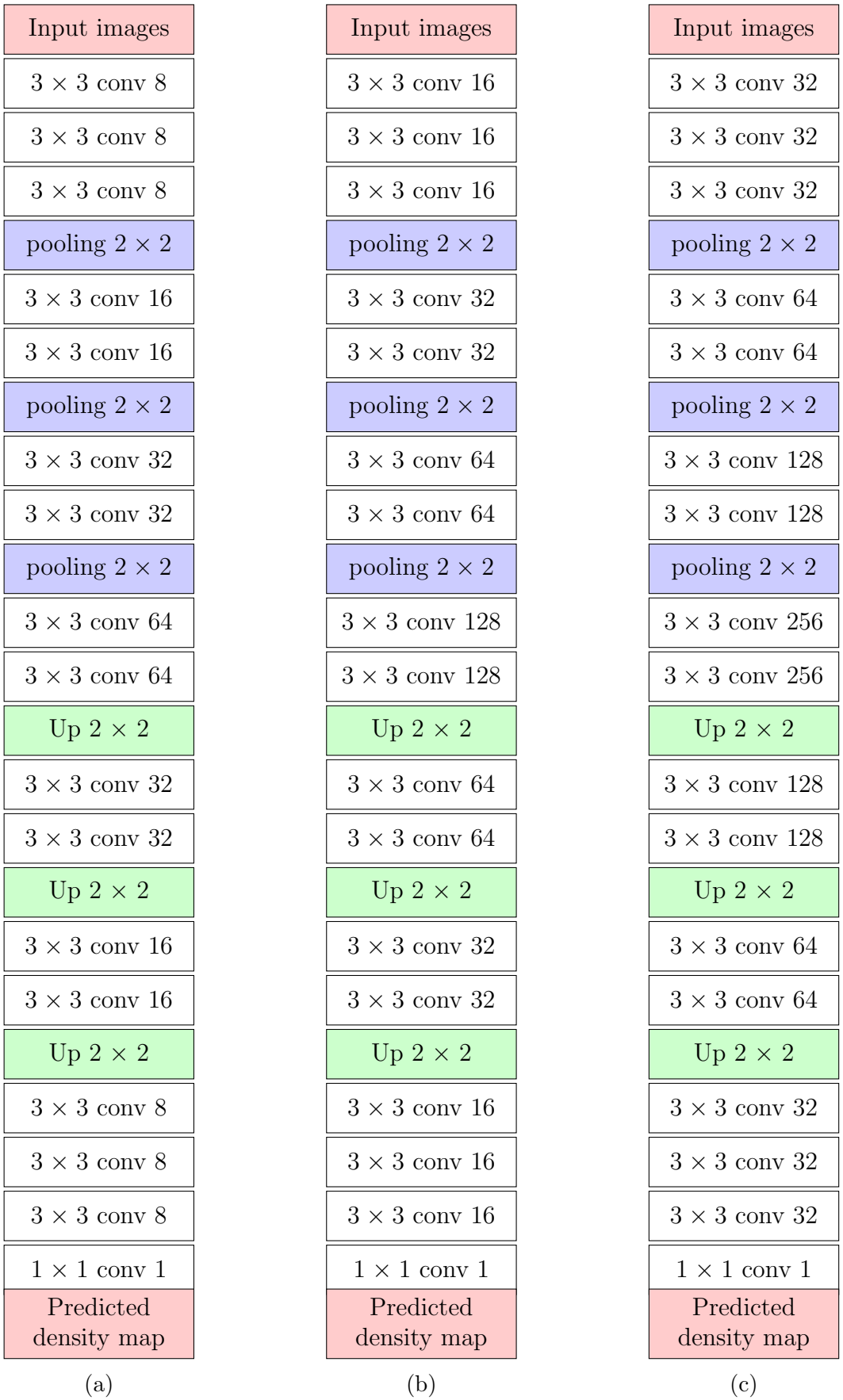



Figure 4.5: The proposed backbone network for reducing number of model parameters consisting of 3 versions, (a) Model B-1, (b) Model B-2, and (c) Model B-3.

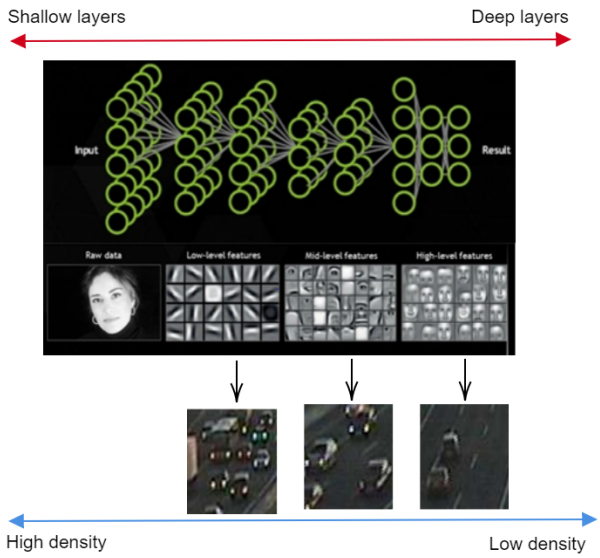



Figure 4.6: Structure of CNNs corresponded to level features and vehicle density [52].

### 4.2.2 Backward skip connection

The design of backward connections is opposite to forward connections, where the features from deeper layers are extracted and concatenated with feature maps from shallower layers, as shown in Figure 4.9. These connections are denoted as Eq. (4.5). It is well known that CNN is a hierarchical network where deep layers require a low-level feature to construct their feature maps. Therefore, the design of the prediction model shown in Figure 4.9 is not deployed in the practical configuration.

$$y_L = F(x_L + y_{N_L+L}) \tag{4.5}$$

To solve this problem, this study proposed techniques for extracting high-level features which are summarized as follows:

- **Master-slave network (MS)** creates two networks called master and slave network. Master networks have a role as a prediction model for predicting density maps and counting the number of vehicles. On the other hand, slave networks have only one role to feed the master networks with high-level feature maps. The slave network is not provided with any backward skip connections from the master network and its predictions are ignored. However, both networks are optimized to minimize losses. The skip connections are denoted as Eq. (4.6).

$$y_L^m = F(x_L^m + y_{N_L+L}^s) \tag{4.6}$$

  where $y_L^m$ and $y_L^s$ are feature maps obtained from $L^{th}$ convolution layer of master and slave networks, respectively. $x_L^m$ is an input of $L^{th}$ Conv layer of master networks. Both master and slave networks have the same network architecture (e.g. the number of Conv layers, kernel sizes, and so on) as shown in Figure 4.10.

- **Batch transfer (BT)** is a training procedure extracting high-level features from predictions that were trained by earlier batches of images. Since CNN is a method

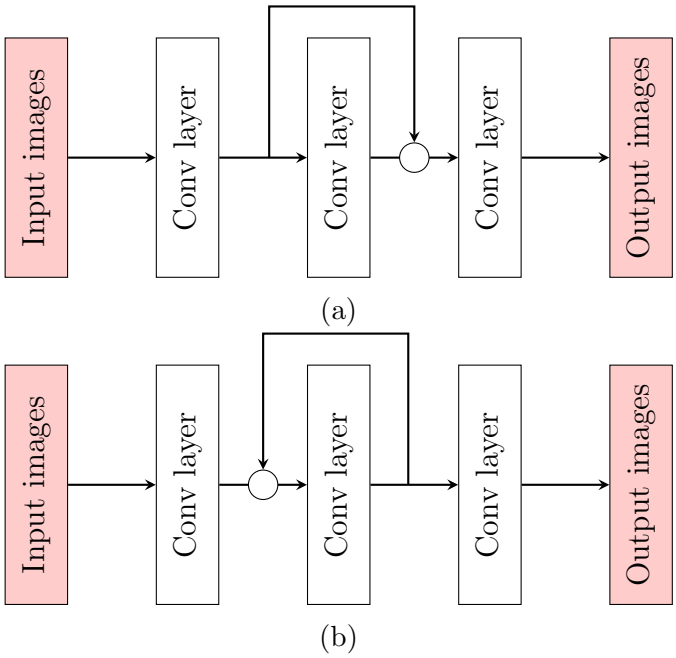



Figure 4.7: CNNs with (a) Forward and (b) Backward connections.

for finding specific characteristics or features from multiple images in different conditions, it is assumed that traffic images have a similar pattern of vehicle density and their feature maps are reused for other batches. The skip connections are denoted as Eq. (4.7).

$$y_L^b = F(x_L^b + y_{N_L+L}^{b-1}) \tag{4.7}$$

where $x_L^b$ is input of $L^{th}$ Conv layer of $b^{th}$ batch trained prediction resulting $y_L^b$. Figure 4.11 shows training procedures in the first three batches, where zero metrics are initially deployed as high-level feature maps in the prediction trained by the first batch of traffic images.

- **Pre-trained map (PT)** uses the same concept of MS but the slave and master networks are separately trained. Since MS has two networks for optimizing their weights, more memory is required twice to predict the number of vehicles. To solve this problem, a transfer learning technique was added to PT. Slave networks have a role as pre-trained networks to calculate high-level features fed to master networks. Figure 4.12 shows the structure of the backbone with PT, where F1, F2, and F3 represent pre-trained feature maps for concatenating with low-level feature maps of master networks.

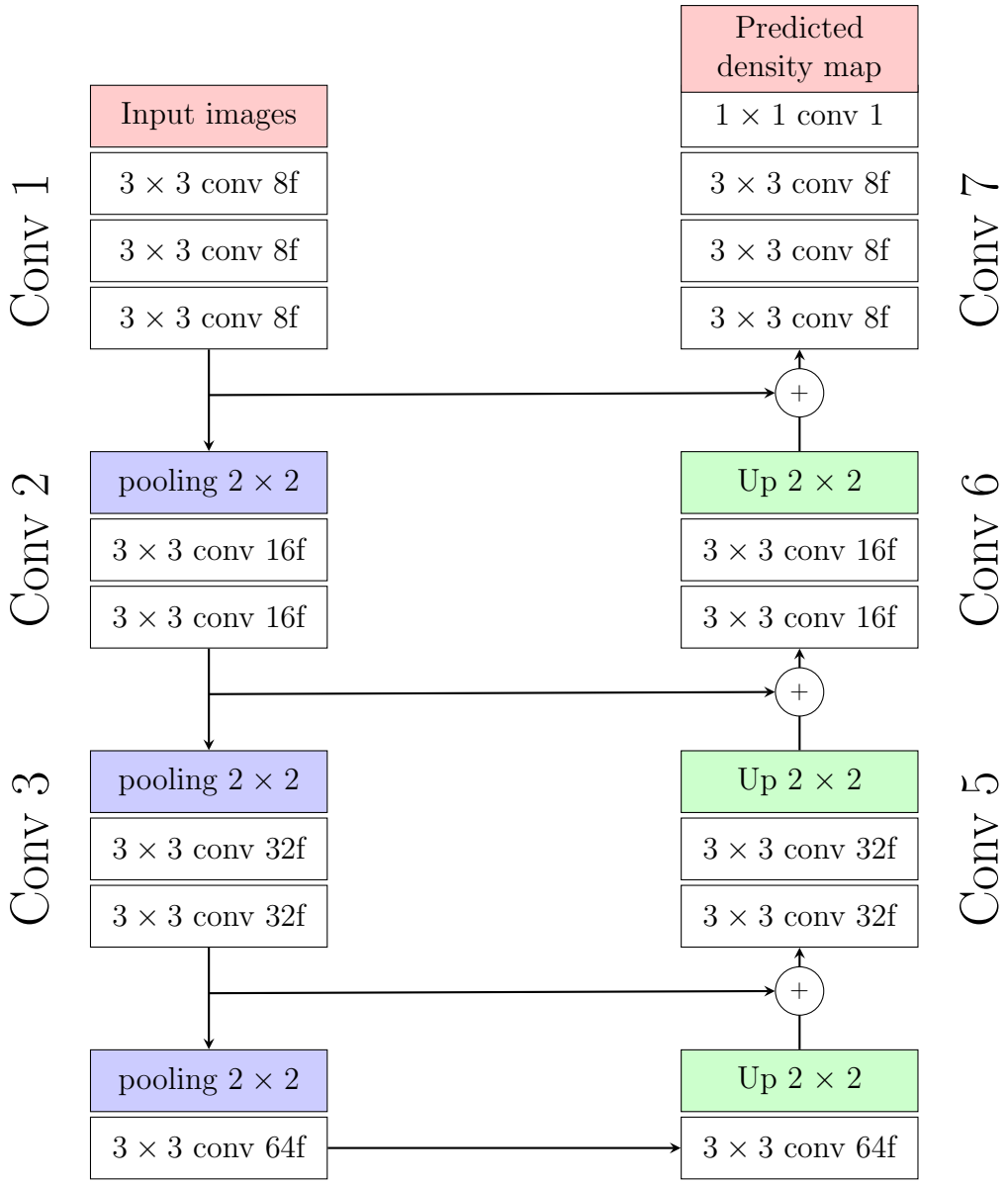



Figure 4.8: The proposed backbone network with forward connections.

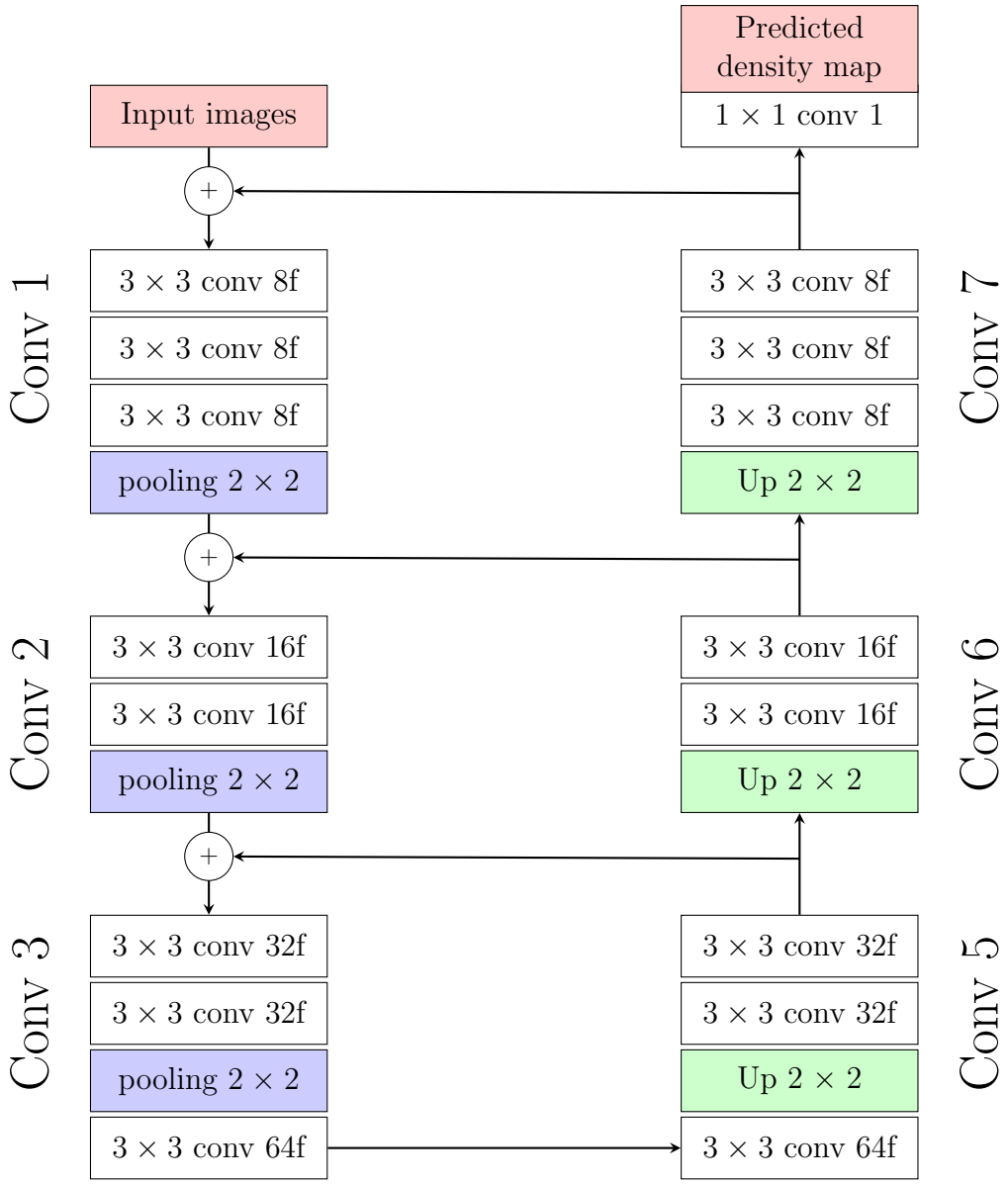



Figure 4.9: The proposed backbone network with backward connections.

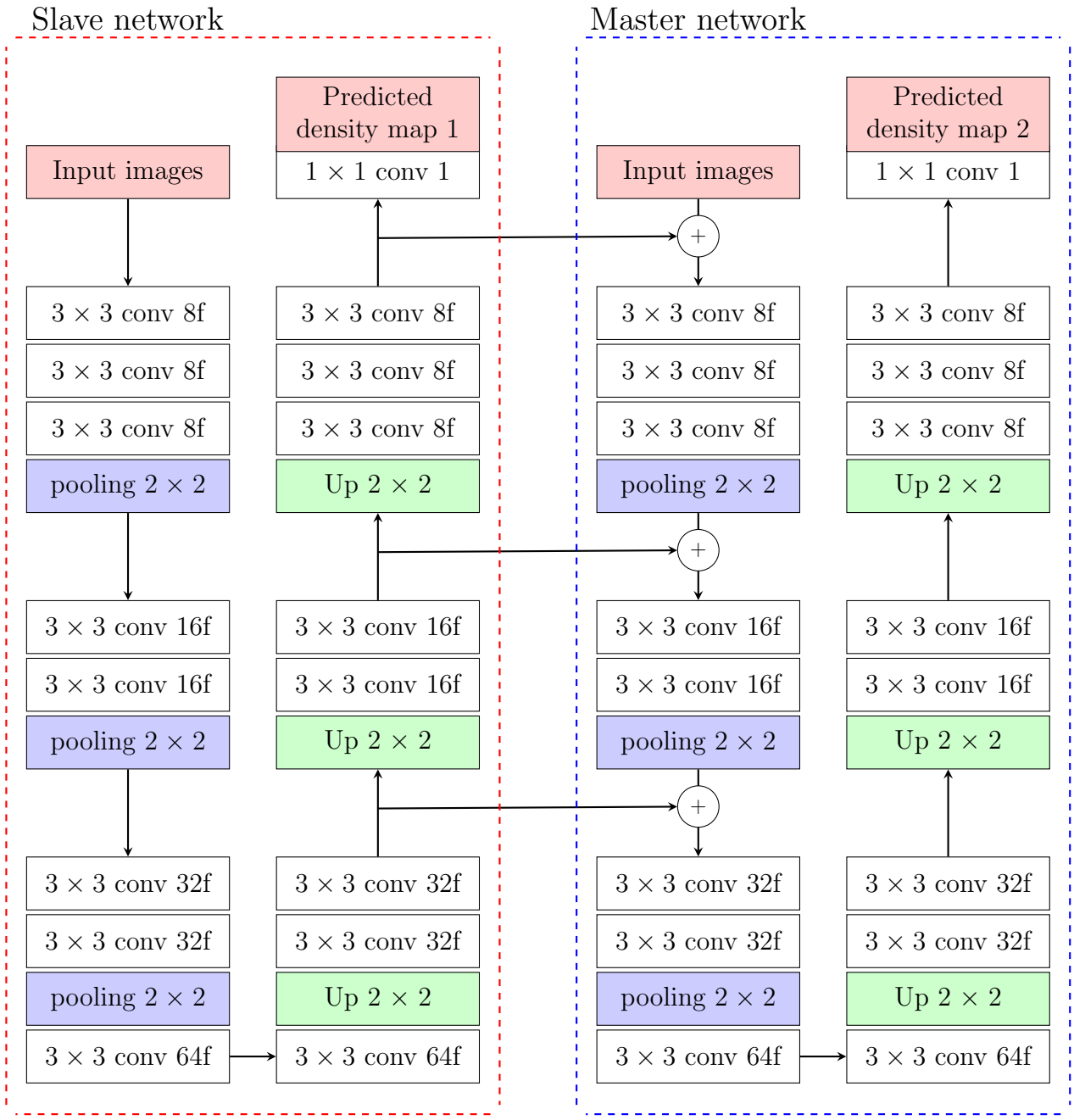



Figure 4.10: The proposed backbone network via slave and master networks where the predicted density map 1 are used for weight optimization in a slave network, while counting results are obtained from the predicted density map 2.

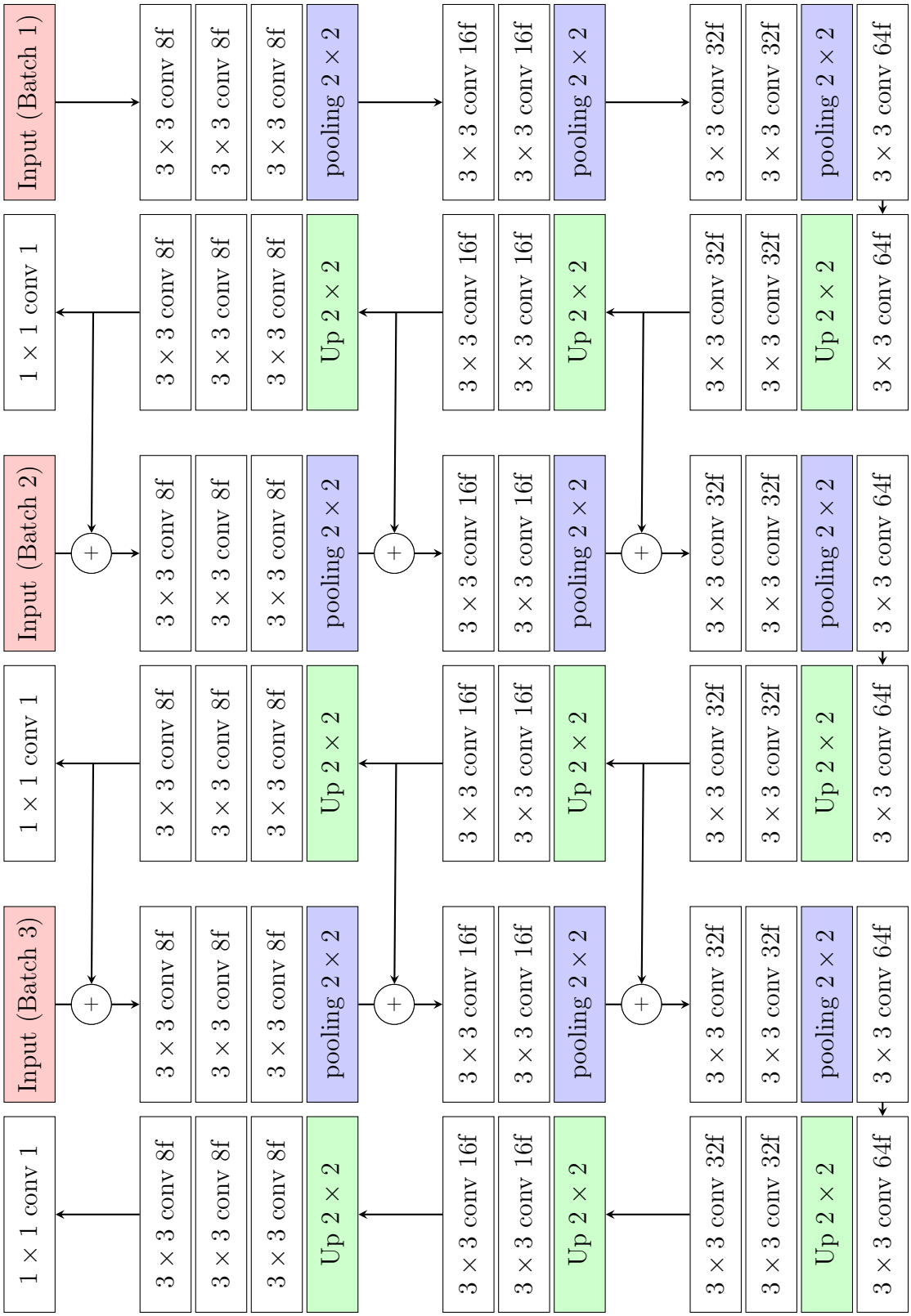



Figure 4.11: The proposed backbone network with batch transfer trained by the first three batches of images.

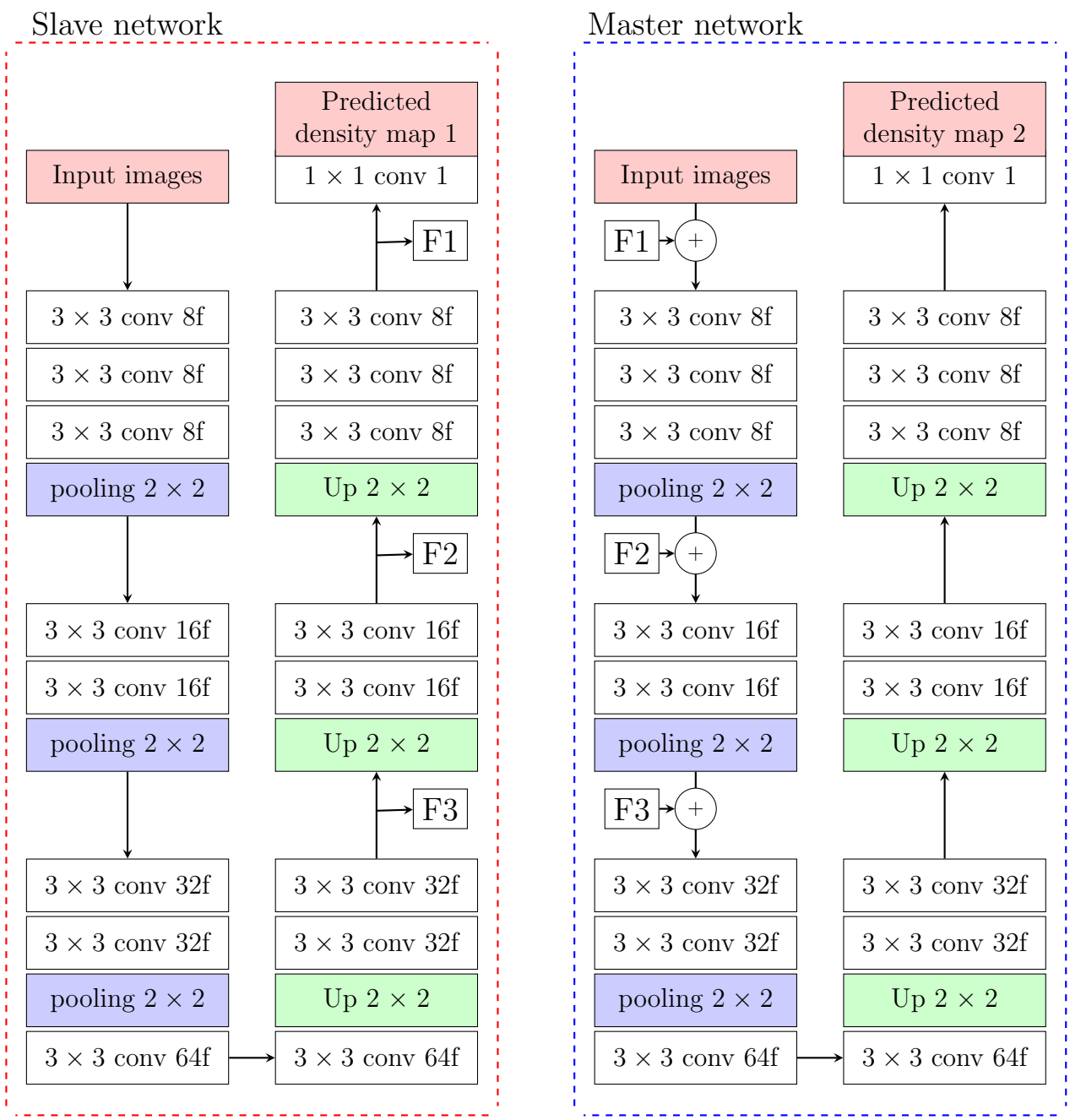



Figure 4.12: The proposed backbone network with Pre-train map where F1, F2, and F3 are pretrained feature maps from a slave network.

## 4.3 Network architecture with dilated convolutions

In general, the structure of the prediction model usually includes pooling and Up-sampling layers to increase the size of the receptive field. However, the previous study [46] proved that resizing feature maps, with pooling and up-sampling layers cause an error in object counting calculated in Eq. (3.7) because valuable information might be lost. Then, the Conv layers should only be employed in the prediction model without resizing feature maps. As a large number of Conv layers, it will require more model parameters for weight optimization. To solve this problem, pooling and up-sampling layers were replaced with dilated convolutions. The dilation convolution is obtained from Eq. (4.8).

$$y(m,n) = \sum_{i=1}^{M}\sum_{j=1}^{N} x(m + r \times i, n + r \times j)w(i,j) \tag{4.8}$$

where $(m,n)$ is the vertical and horizontal coordinates of the feature map and $y(m,n)$ is the output of a dilated Conv layer from their input features $(x(m,n))$ and weights $(w(m,n))$ with the height and width of $M$ and $N$, respectively. The parameter $r$ is the dilation rate, where the normal convolution are assigned with $r = 1$. The dilated Conv layer can increase the size of the receptive field exponentially without resizing. In addition, the experiment from [46] has proved that the counting accuracy is improved by increasing the quality of the predicted density map with dilated convolutions. Based on our experiment, a backbone network with a dilated rate = 2 is selected because it gives the most optimal performance of the density map prediction. To keep a feature map resolution, all Conv layers use zero-padding to maintain the previous size.

Let Conv (3, 16, 1) is represented as a Conv layer with $3 \times 3$ kernel sizes, the number of filter = 16, and a dilated rate = 1. With the advantage of the dilated convolution, network architectures with backward connections can be modified by adding dilated convolution from section MS, BT, and PT as shown in Figures 4.13, 4.14, and 4.15, respectively, where dilated convolutions are represented as yellow rectangular boxes.

## 4.4 Summary

This chapter describes the network architectures of the prediction model in this study. First of all, our backbone network was designed from the state-of-the-art crowd counting where their kernels are changed to a stack of Conv layers $(3 \times 3)$ for reducing the number of model parameters. To increase prediction performance, we propose a prediction model with a skip connection consisting of forward and backward connections. The prediction model with forward connections employs concatenation between low-level and high-level feature maps with the same dimensions. On the other hand, models with backward connections are a reverse version of forward connection consisting of three versions. First, another network was designed without concatenated feature maps and transfer its high-level feature map to a master network. Second, high-level feature maps are transferred from previously trained batch networks by initializing concatenated feature maps in the first batch as zero matrices. Third, the slave network is also employed but it has a role as pre-trained networks to provide its high-level feature map for a master network. In the final version, dilated convolutions were introduced to recover information loss from pooling and up-sampling layers while keeping the size of the receptive field.

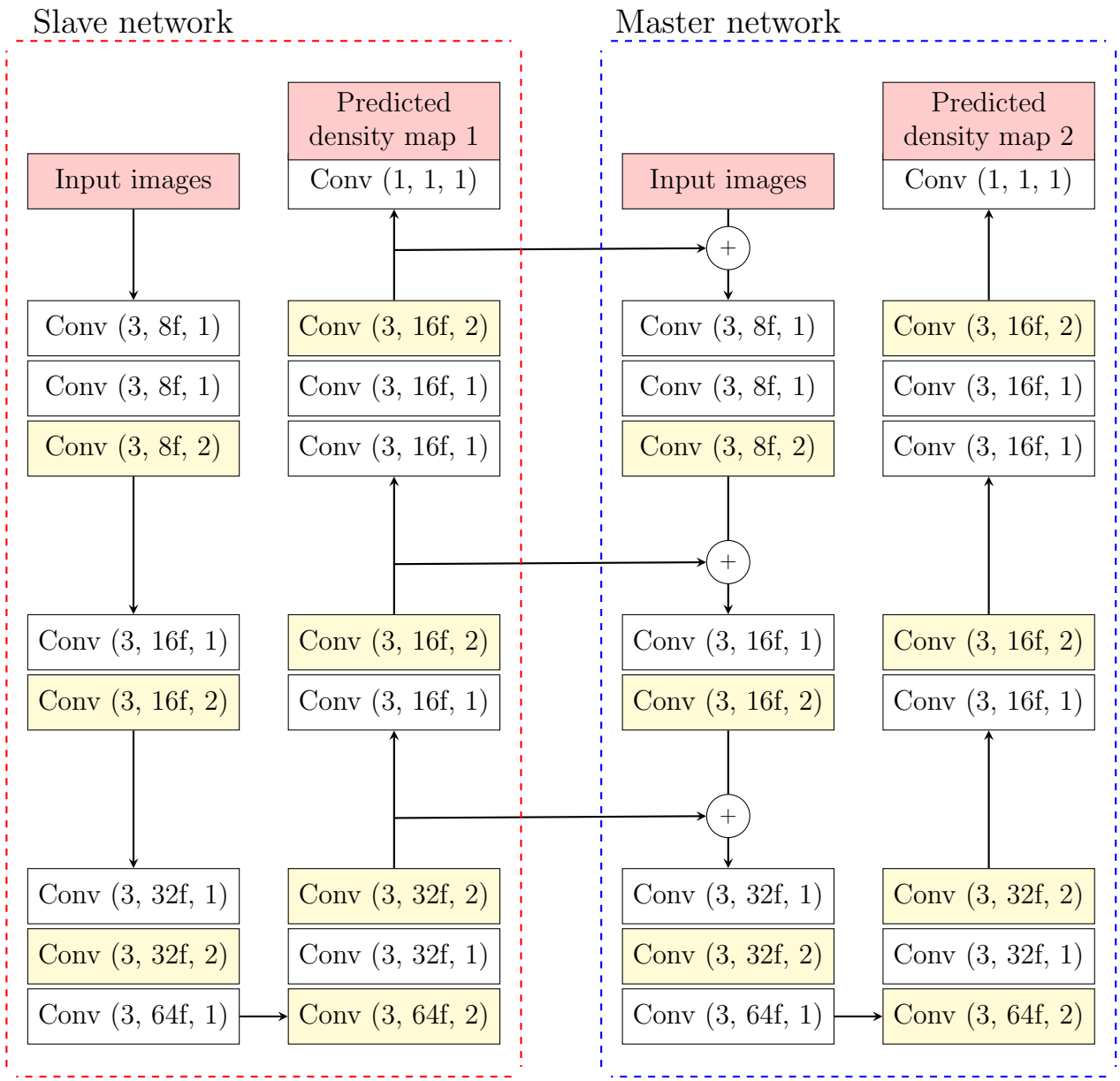



Figure 4.13: The proposed backbone network with slave and master networks and dilated convolutions where predicted density map 1 are used for weight optimization in a slave network, while counting results are obtained from a predicted density map 2.

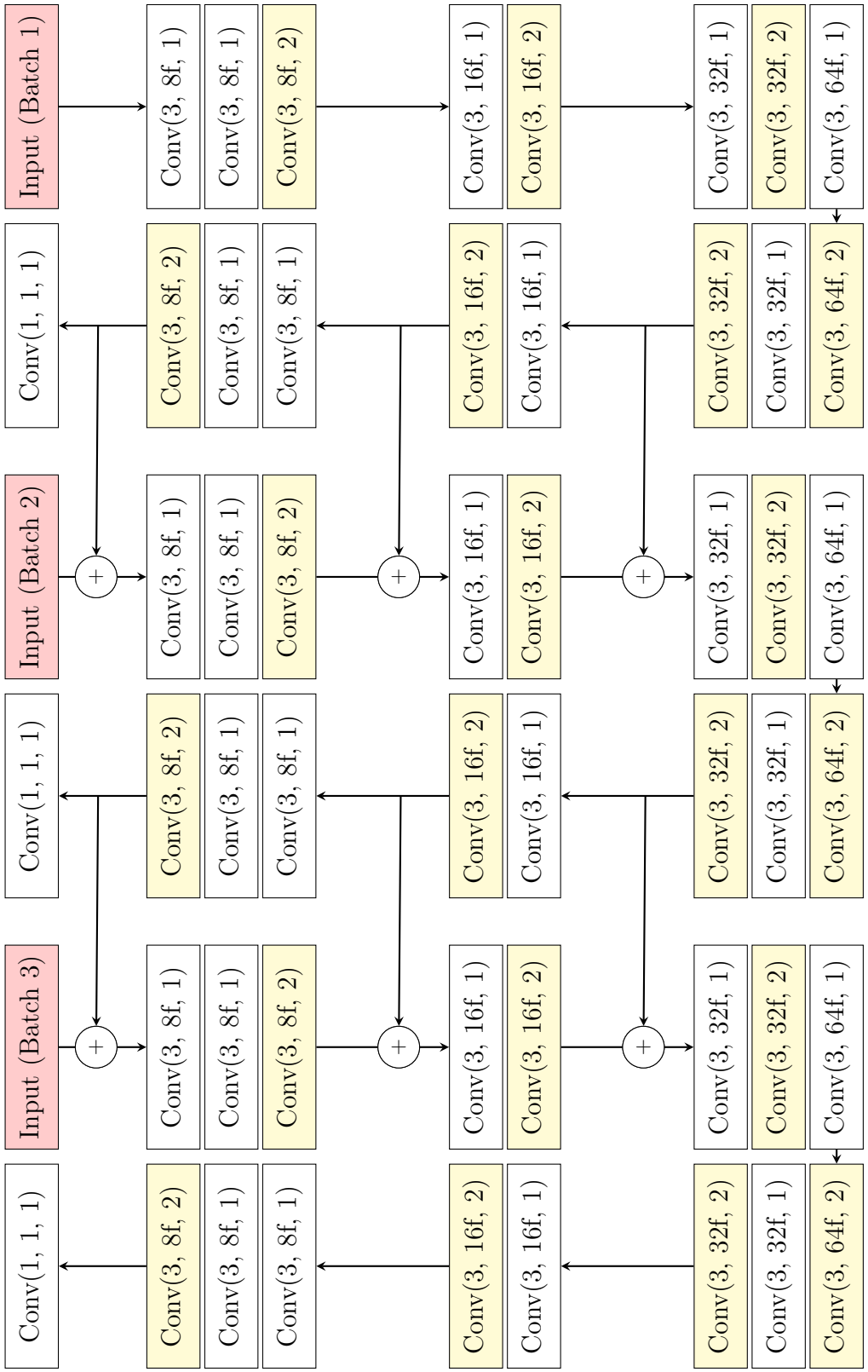



Figure 4.14: The proposed backbone network with batch transfer and dilated convolutions trained by the first three batches of images.

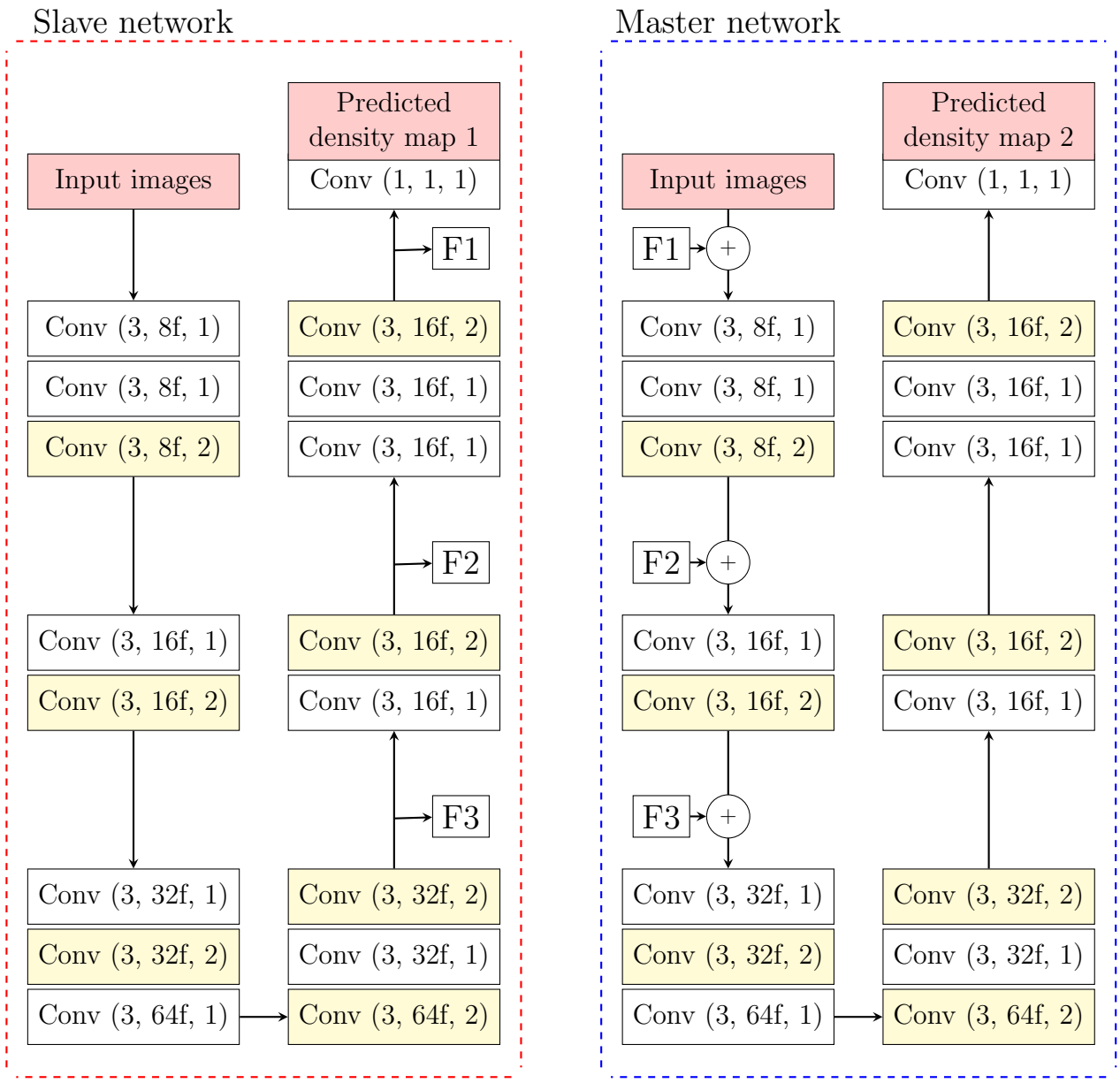



Figure 4.15: The proposed backbone network with Pre-train maps and dilated convolutions where F1, F2, and F3 are pretrained feature maps from a slave network.

# Chapter 5

# Experimental Results

This chapter describes empirical results where the proposed methods and related studies were evaluated. The chapter was divided into 4 sections, experimental setup, vehicle counting evaluation of the prediction model, and analysis on a predicted density map. First, the experimental setup gives information about pre-processing factors related to the evaluation of the prediction model. Second, the empirical results were analyzed in different aspects. Third, the factor on the quality of the predicted density map was discussed. In the last section, the proposed prediction model was evaluated in other datasets, in addition to vehicle density estimation. Their descriptions are shown as follows:

## 5.1 Experimental setup

### 5.1.1 Hardware and software specification

Since the aim of my study is to examine the single-column network requiring only one CNN in a prediction model, where computation cost is concerned. Therefore, PC specification computing and optimizing trainable parameters should be described to compare time complexity and memory consumption. The hardware specification is summarized as follows:

- CPU: Intel Xeon E5-2680v2 10Core
- GPU: NVIDIA Tesla K40
- Memory: 64GB

In addition, the prediction models were implemented by Python 3.5.6. Figure 5.1 shows that the models implemented with Pytorch, Tensorflow, and Keras have a different training time. In our experiment, all prediction models were implemented by Tensorflow 1.10.0 because less computational cost is required than the other two libraries.

### 5.1.2 Traffic image dataset

As mentioned in Chapter 1, the traffic analysis based on computer vision is prominent for traffic monitoring systems. Then, several traffic datasets for vehicle detection were introduced to evaluate the proposed method in various applications in ITS. This subsection

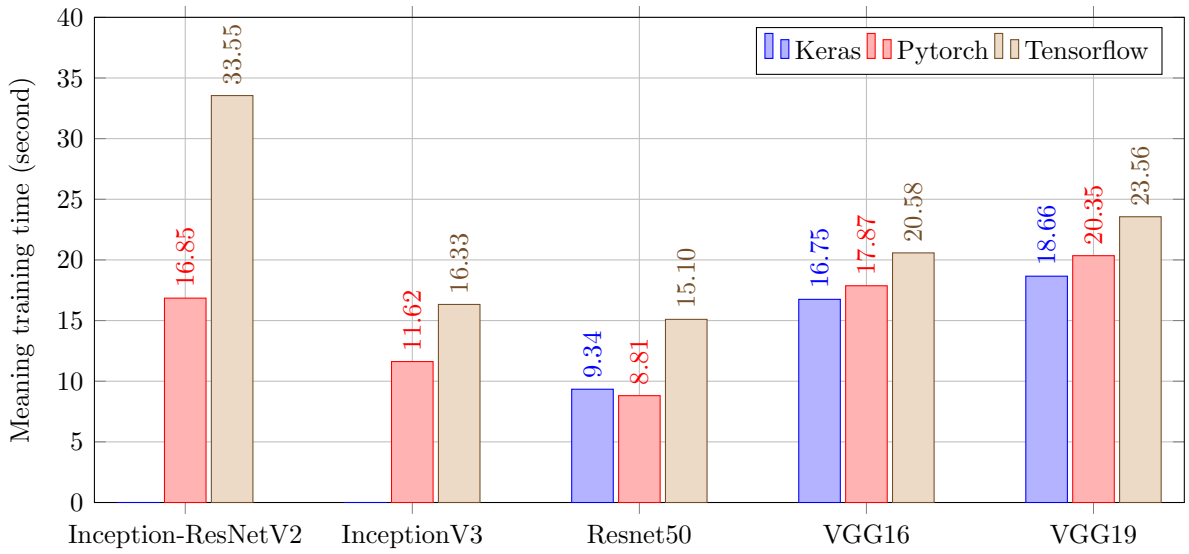



Figure 5.1: Average training time from well known CNNs implemented with Pytorch, Tensorflow, and Keras [67].

provides the list of benchmark datasets for vehicle density estimation. their overviews and characteristics are briefly introduced as follows:

- **PASCAL VOC dataset** [22] is one of the well-known dataset for object detection, including vehicle detection. The principal challenge is to evaluate the detection algorithm for each object category, where images in the dataset can be categorized into 20 classes. object annotation is provided for the twenty classes. all images are annotated with bounding boxes for every instance of the twenty classes for the classification and detection challenges. In addition, their attributes (e.g. 'orientation', 'occluded', 'truncated', and 'difficult) are specified with bounding boxes.

  Objects with the vehicle class have 7 instances. The images containing vehicle classes were captured in different camera viewpoints as shown in Figure 5.2. However, their camera viewpoints are different from traffic videos recorded by a surveillance system.

- **KITTI dataset** [24] provides traffic videos recorded by cameras installed on vehicles. This dataset is deployed for object detection, and other tasks. Figure 5.3 shows example images from this dataset, where their traffic images are categorized into city, residential, and road areas shown in Figures 5.3a, 5.3b, and 5.3c, respectively.

  The object class is denoted as 'vehicle', 'Truck', 'Pedestrian', 'Person (sitting)', 'Cyclist', 'Tram' and 'Misc' (e.g., Trailers, Segways, and so on). For each image, the translation and rotation are provided in 3D, while the other two angles are assumed to be close to zero. Furthermore, vehicle density and truncation are specified.

- **UA-DETRAC Benchmark Suite** [89] contains traffic images recorded by 100 surveillance cameras with high resolutions. Traffic videos have been labeled their detailed attributes (e.g. vehicle category, weather, occlusion ratio, and scale) in the dataset. In each scene, vehicles are marked with their locations and attributes as shown in Figure 5.4. Besides, ignorance regions are specified. These regions contain

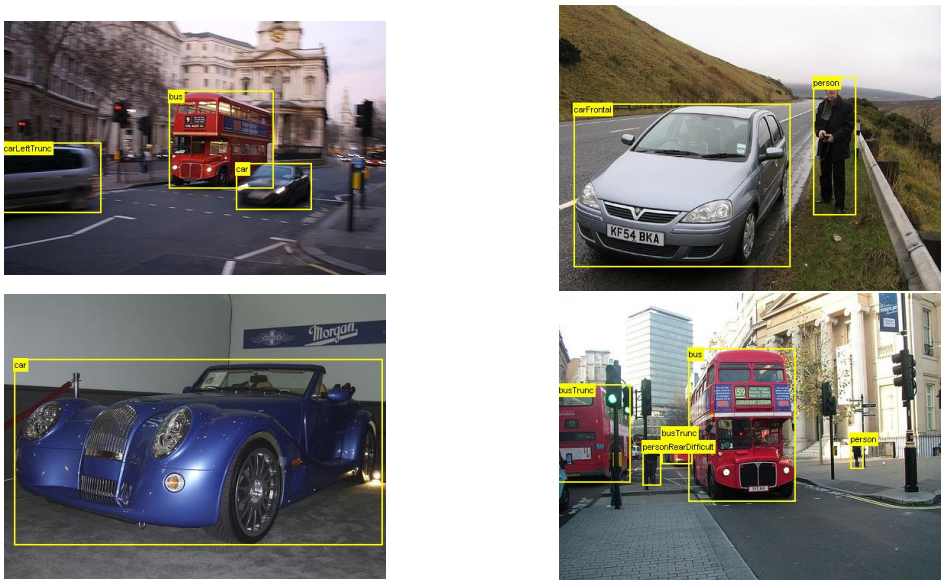



Figure 5.2: The example of images from PASCAL VOC dataset [22] containing objects with vehicle classes and complete annotations at various viewpoints.

(a) (b) (c)

Figure 5.3: The examples of images from the KITTI dataset [24] recorded at (a) city, (b) residential, and (c) road areas.

vehicles that were not annotated or detected because they have very small sizes illustrated in Figure 5.4 as dark regions.

The real-world part of the UA-DETRAC dataset is captured under three occlusion status as fully visible, partially occluded by other vehicles and partially occluded by background, and different degrees of truncation as vehicle parts outside the image. To achieve a training distribution, the multi-scale vehicles including small (0 - 50 pixels), medium (50 - 150 pixels), and large (more than 150 pixels) are extracted to train the predicition model.


Figure 5.4: The examples of images from UA-DETRAC Benchmark Suite [89].

- **TRANCOS dataset**[64] presents the dataset of traffic images with extremely overlapping vehicles. Figure 5.5 depicts the examples of traffic images in this dataset. These images show how challenging the problem if extremely overlapping vehicles are captured by a traffic surveillance camera.

  The dataset consists of 1244 images. They have been acquired from a selection of public traffic surveillance cameras provided by the Directorate General of Traffic (DGT) of the Government of Spain. The cameras selected monitor different highways in Madrid, which typically presents heavy traffic congestion.

To select a suitable dataset, their images should have the same characteristic of traffic surveillance images in practical circumstances. Typically, public authorities install cameras in high positions (over the road) to capture whole traffic scenarios as large as possible. It implies that traffic images from PASCAL VOC [22] and KITTI datasets[24] are not suitable for VIVDS. As mentioned in section 1.2, Most of the traffic videos have low resolutions and frame rates, where traffic images from UA-DETRAC Benchmark Suite [89] hardly employed in practical circumstances. Therefore, TRANCOS dataset is selected for the evaluation of vehicle density estimation in this study.

## 5.2 Vehicle counting evaluation of prediction model

This section describes the performance of the proposed prediction models mentioned in Chapter 4. Each model is evaluated in the following subsections. Since this study is

Figure 5.5: The examples of traffic images from TRANCOS dataset [64].

related to VIVDS in the traffic light control system, their evaluation criteria should be calculated in this experiment. Based on the Departments Approved Product List (APL) [6], the minimum accuracy of each traffic attributes for VIVDS are as follows:

- Density: 95 % (normal weathers) and 90% (adverse weathers).
- Occupancy: 90 % (normal weathers) and 85% (adverse weathers)
- Speed: 90 % (normal weathers) and 85% (adverse weathers)

The counting results are reported within 1 - 10 seconds. vehicle density accuracy was calculated in Eqs. (5.1) and (5.2), where $VT_{ln_i}$ and $VT^*_{ln_i}$ are total vehicle density in $i^{th}$ lane from prediction models and ground truths, respectively. $VA_{ln_i}$ and $VA$ are vehicle density accuracy in the $i^{th}$ lane and its average values, respectively. $N$ is the total number of lanes in the traffic images. Since most related works are evaluated by MSE and MAE calculated in Eqs. (3.8) and (3.9), respectively, their vehicle density accuracy was obtained by using Eq. (5.3) where $M$ is the number of tested images.

$$VA_{ln_i} = 100 - \frac{|VT_{ln_i} - VT^*_{ln_i}|}{VT^*_{ln_i}} \times 100 \tag{5.1}$$

$$VA = \frac{\sum_{i=1}^{N} VA_{ln_i}}{N} \tag{5.2}$$

$$VA = 100 - \frac{MAE \times M}{VT^*} \times 100 \tag{5.3}$$

As observed in the evaluation metric, the estimation network was only evaluated in image-level performance (MSE and MAE). Even though point-level performance (Recall and Precision) in detection-based method provides more valuable information, it was not taken into account as following reasons:

- With low resolutions of targets and images, a predicted location from the density map is difficulty estimated directly from the centroid matrix (H) in Chapter 3, due to a high crowd density.
- In case where the density pf target is high, every target is generally located in most parts of ROI. Inaccurate target localization is not extremely effective for a regression-based approach in the practical application.

### 5.2.1 Ablations on the backbone network

The backbone network with various modifications is evaluated in this subsection. As mentioned in section 4.1, backbone networks are designed from one stack of MCNN [95] resulting Model A-1, A-2, and A-3 as shown in Figure 4.3. Besides, their kernel sizes were modified by using $3 \times 3$ Conv layers resulting Model B-1, B-2, and B-3 as shown in Figure 4.5. Their evaluation performance is shown in Table 5.1. It shows that prediction models using stacks of $3 \times 3$ Conv layers have higher average accuracy. Varying the number of trainable parameters reveals that models A-2 and B-2 has the highest accuracy. Even though models A-3 and B-3 have a larger number of trainable parameters, their performances are reduced. This error is caused by an overfitting problem. Therefore, model B-2 will be employed for the rest of the experiments. For time complexity, training, and testing time for each model depends on the number of trainable parameters. However, their time consumption might not satisfy the minimum requirements of VIVDS because our hardware specification is different from the practical configuration. This information will be used for comparison with other models in the next subsection.

| Model name | No. Trainable parameter | MAE | VA (%) | Training time (s) | Testing time (s) |
|---|---|---|---|---|---|
| A-1 | 138,833 | 9.40 | 76.24 | 0.096 | 0.07 |
| A-2 | 552,609 | 6.29 | 84.09 | 0.16 | 0.07 |
| A-3 | 2,204,993 | 8.12 | 79.47 | 0.33 | 0.07 |
| B-1 | 111,185 | 5.37 | 86.43 | 0.089 | 0.06 |
| B-2 | 443,553 | 4.13 | 89.55 | 0.12 | 0.07 |
| B-3 | 1,771,841 | 5.46 | 86.2 | 0.21 | 0.06 |

Table 5.1: Vehicle counting accuracy by using prediction models with reducing kernels sizes

Figure 5.6 illustrates the density error ($D_v$) from proposed prediction models: models A-1, A-2, and A-3 (Figures 5.6a, 5.6a, and 5.6c respectively). The density error is calculated by using Eq. (5.4), where $y$ and $\hat{y}$ are actual and predicted counts, respectively. The positive and negative values of $D_v$ indicate the quality of over-counting and under-counting errors, respectively. They show that the density error, especially in under-counting error, is reduced when more parameters are extracted. However, the predicted count is decreased. On the other hand, the density error from models B-1, B-2, and B-3, shown in Figures 5.7a, 5.7b, and 5.7c respectively, are effective to reduce under-counting error. However, the over-counting error was not solved by their models, especially in a large number of vehicles ($> 50$ vehicles).

$$D_v(y, \hat{y}) = \hat{y} - y \tag{5.4}$$

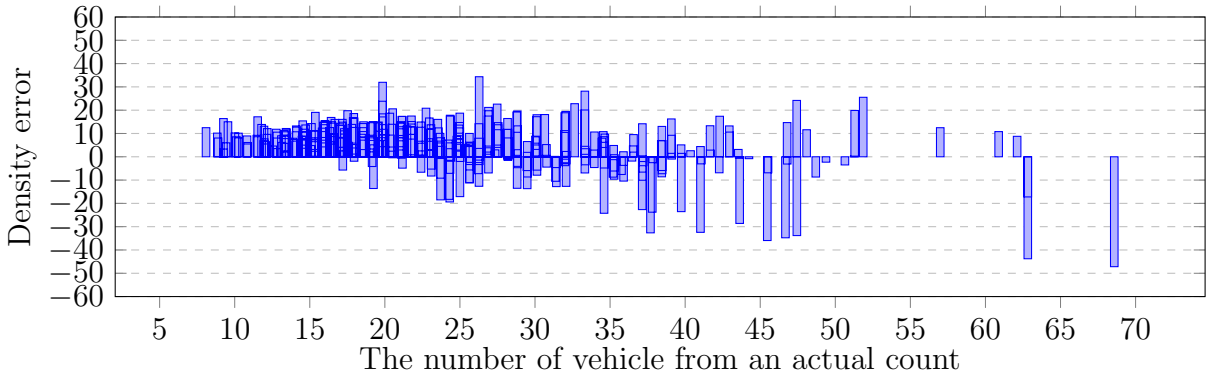
### 5.2.2 Ablations on the skip connection and dilated convolution

This subsection describes the effect of skip connections and dilated convolutions. As mentioned in section 4.2, prediction models with skip connections were categorized into forward and backward connections. In addition, pooling and Up-sampling layers were replaced with dilated convolutions to avoid information loss by resizing images in Section 4.3. Table 5.2 shows the counting accuracy of model B-2 with and without skip connections. The experimental results show that models with skip connections increase density accuracy, especially in models with backward connections (MS, BT, and PT). Even though MS has a high accuracy than FC, MS requires more memory space for trainable parameters. The performance from the model with BT is not significantly improved. It proves that the characteristic among traffic images are varied to each other. On the other hand, dilated convolutions can help to increase VA around 1% from backward connections where their benefits are effective to the quality of density maps discussed in Section 5.3. Since PT with dilated convolution has the best counting accuracy in this research, it will be compared with other prediction models in the next subsection.

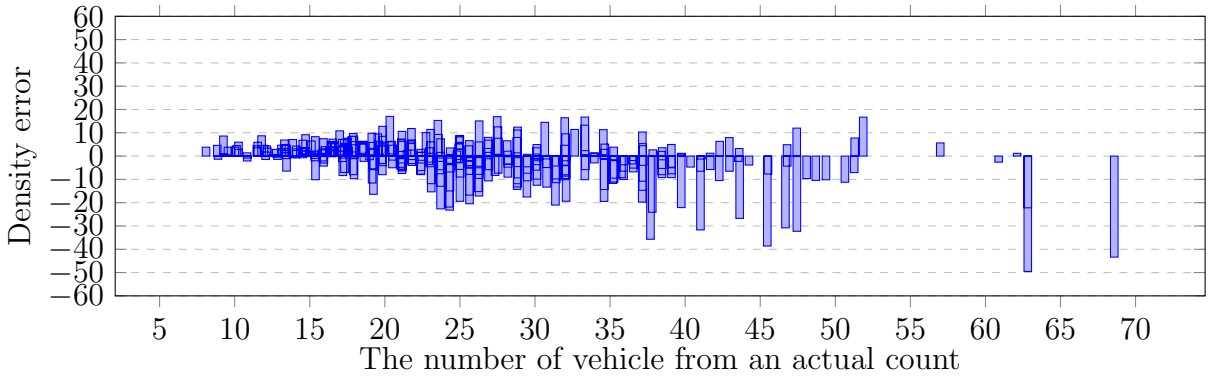
In terms of the density error calculated by Eq. (5.4), the density error is depicted in Figures 5.8a, 5.8b, 5.8c, and 5.8d from model B-2 with FC, MS, BT, and PT, respectively. By adding skip connections, the predicted count is closer to the actual counts than model B-2 results shown in Figure 5.7b. In the skip connection, It shows that FC and MS suffer from under-counting errors, while PT and BT have the same problem on over-counting and under-counting errors. On the other hand, dilated convolutions are effective to reduce under-counting error, especially in high vehicle density as shown in Figure 5.9. However, it can be seen that the density error from prediction models with dilated convolution (MS and BT) are increased on traffic images containing $20 - 40$ vehicles as shown in Figures 5.9a and 5.9b respectively. It shows that models with a large number of parameters from MS and BT are hard to be optimized by dilated convolutions. Meanwhile, PT with dilated convolutions is ineffective for this issue as shown in Figure 5.9c.

| Model name | No. Trainable parameter | MAE | VA (%) | Training time (s) | Testing time (s) |
|---|---|---|---|---|---|
| B-2 | 443,553 | 4.13 | 89.55 | 0.12 | 0.07 |
| B-2 (FC) | 491,937 | 3.70 | 90.65 | 0.12 | 0.09 |
| B-2 (MS) | 983,874 | 3.38 | 91.46 | 0.54 | 0.21 |
| B-2 (BT) | 491,937 | 3.98 | 89.96 | 0.36 | 0.13 |
| B-2 (PT) | 491,937 | 2.99 | 92.47 | 0.33 | 0.11 |
| B-2 (MS) with dilated Conv | 983,874 | 2.46 | 93.79 | 0.55 | 0.20 |
| B-2 (BT) with dilated Conv | 491,937 | 4.96 | 87.47 | 0.35 | 0.13 |
| B-2 (PT) with dilated Conv | 491,937 | 3.01 | 92.40 | 0.34 | 0.11 |

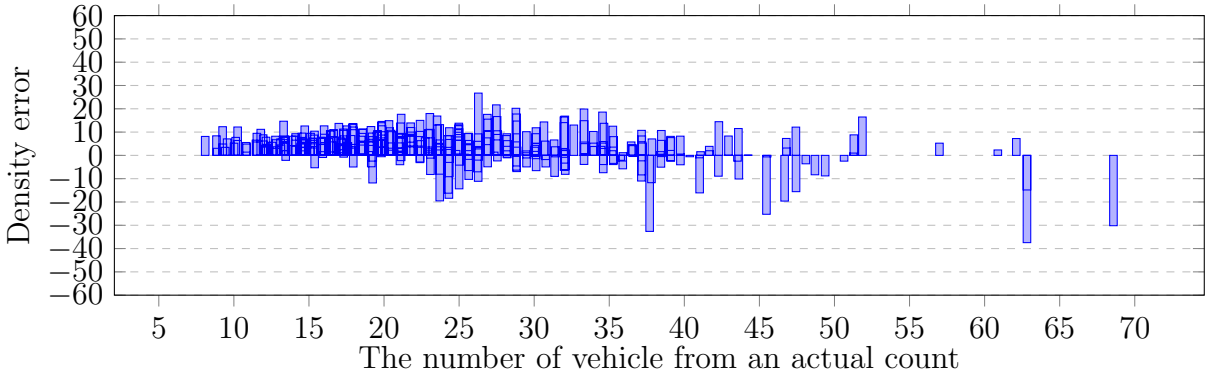
Table 5.2: Vehicle counting accuracy by using prediction models with skip connections consisting models B-2 with forward connections (FC), master-slave network (MS), Batch transfer (BT), and pre-trained map (PT) with and without dilated convolutions.




(a)

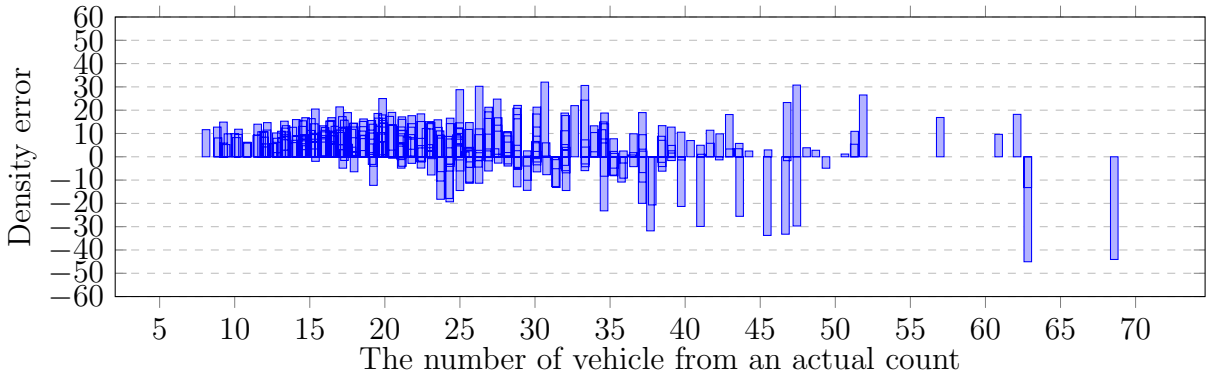



(b)

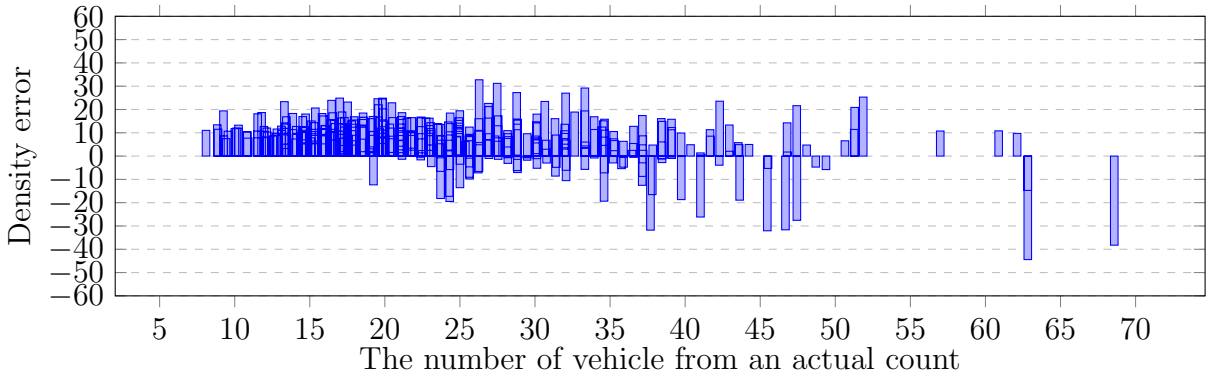



(c)

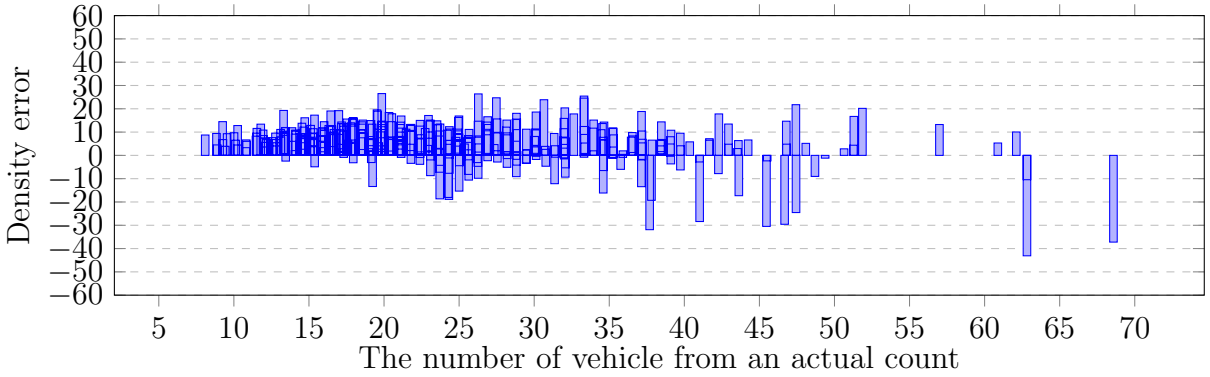
Figure 5.6: The density error from TRANCOS dataset [64] by using (a) model A-1, (b) model A-2, and (c) model A-3.




(a)

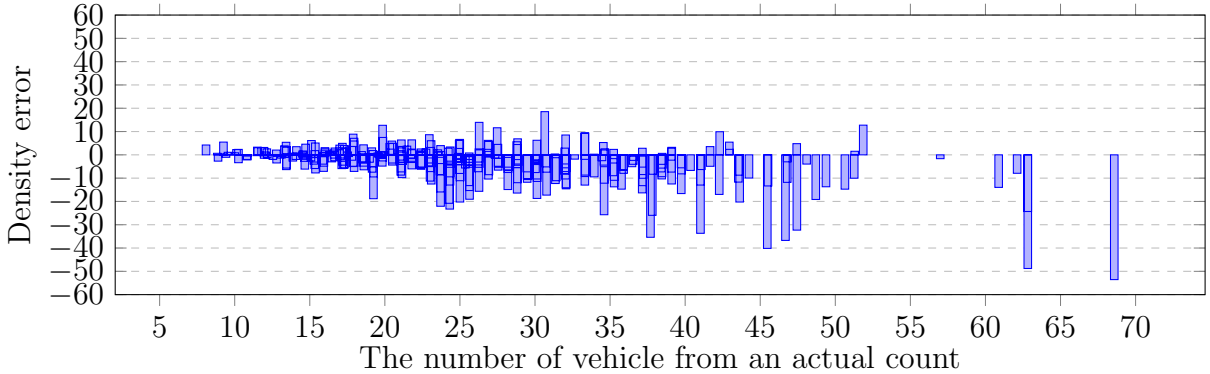



(b)

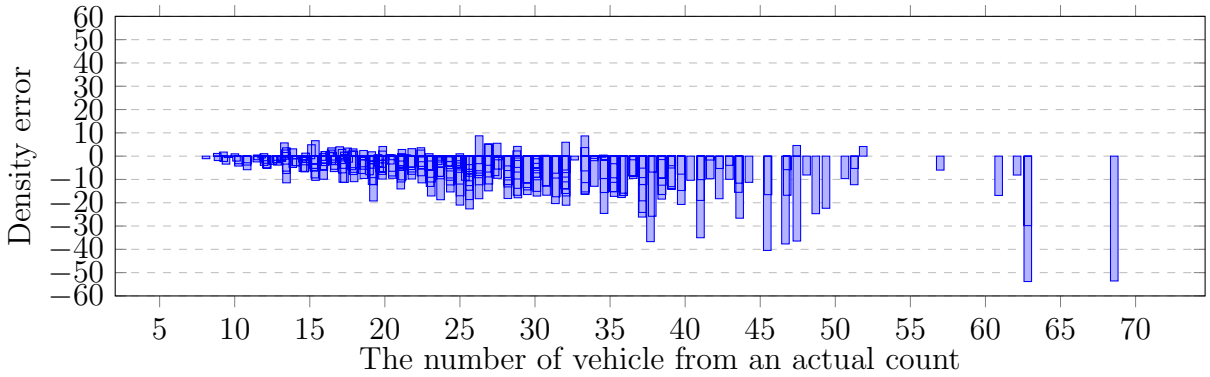



(c)

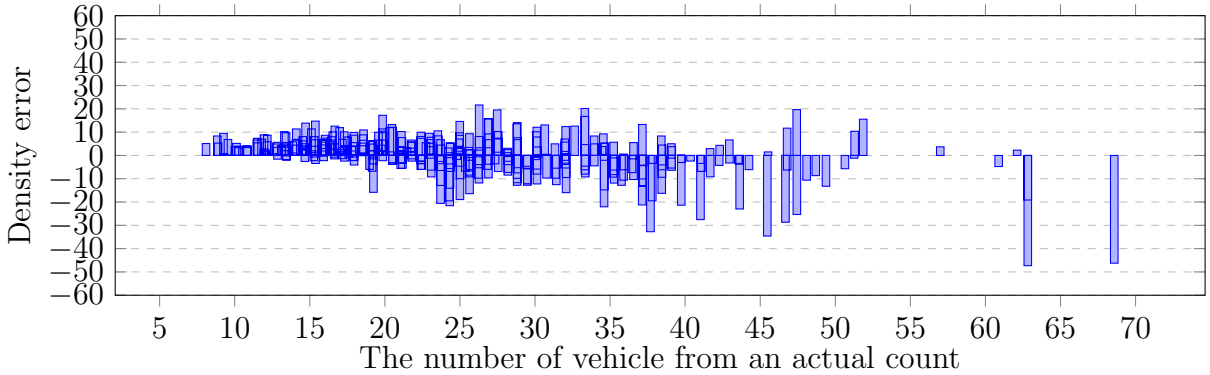
Figure 5.7: The density error from TRANCOS dataset [64] by using (a) model B-1, (b) model B-2, and (c) model B-3.



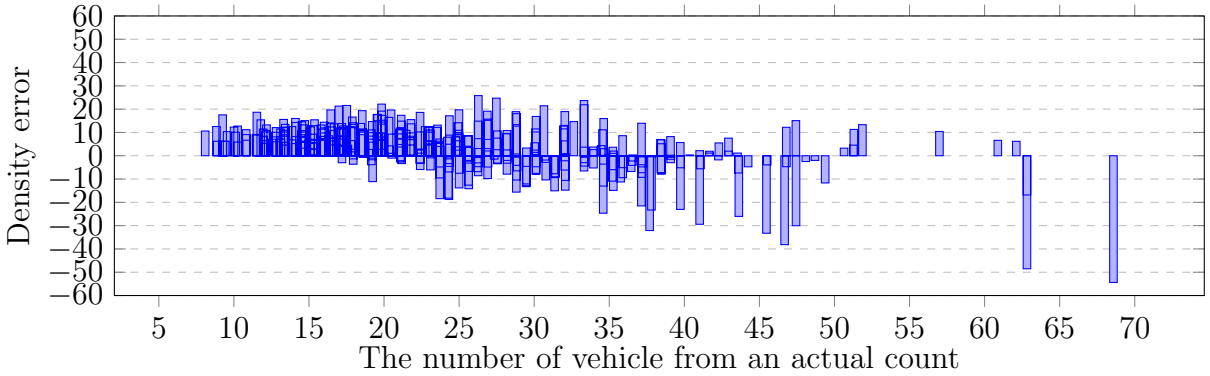

(a)




(b)

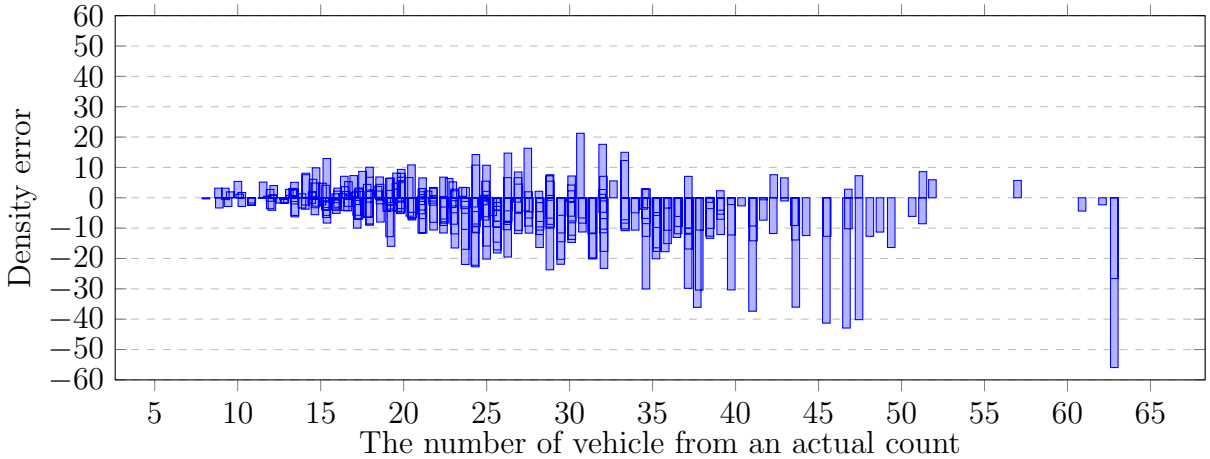



(c)

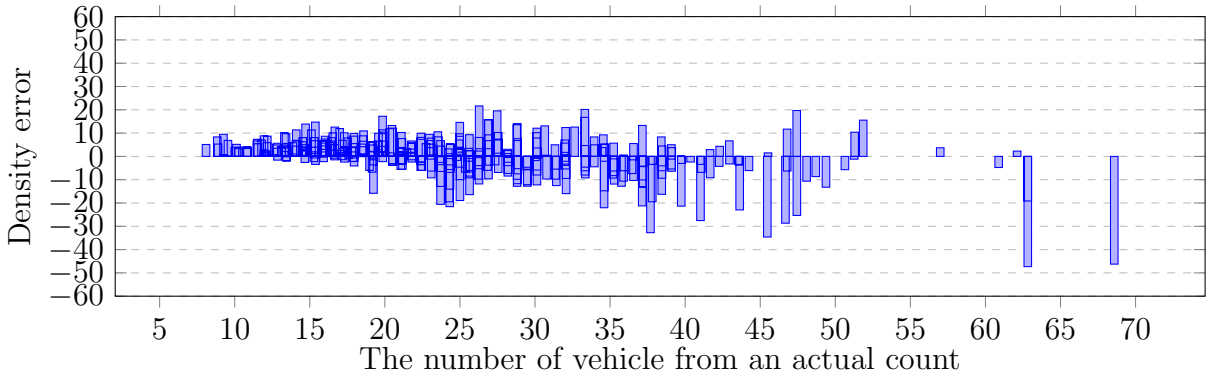



(d)

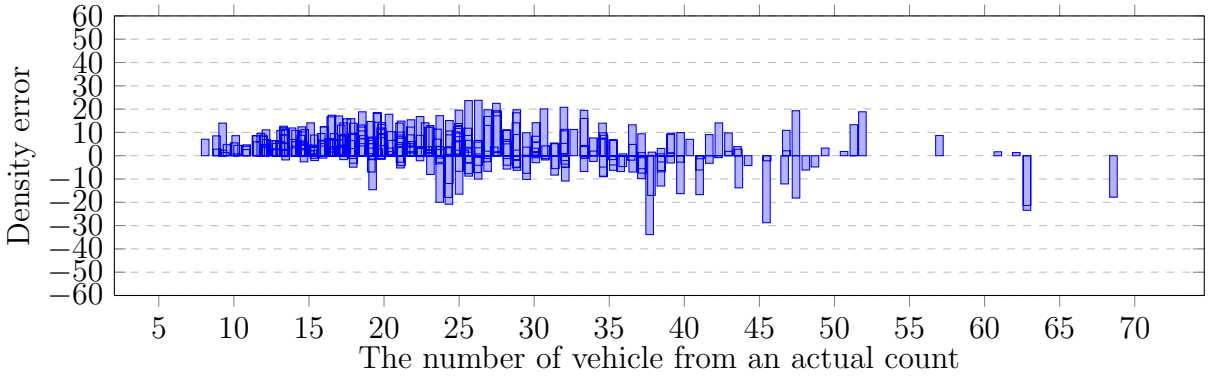
Figure 5.8: The density error from TRANCOS dataset [64] by using (a)FC, (b) MS, (c) BT, and (d) PT.




(a)




(b)




(c)

Figure 5.9: The density error from TRANCOS dataset [64] by using (a) MS, (b) BT, and (c) PT with dilated convolutions.

### 5.2.3 Counting accuracy with related works

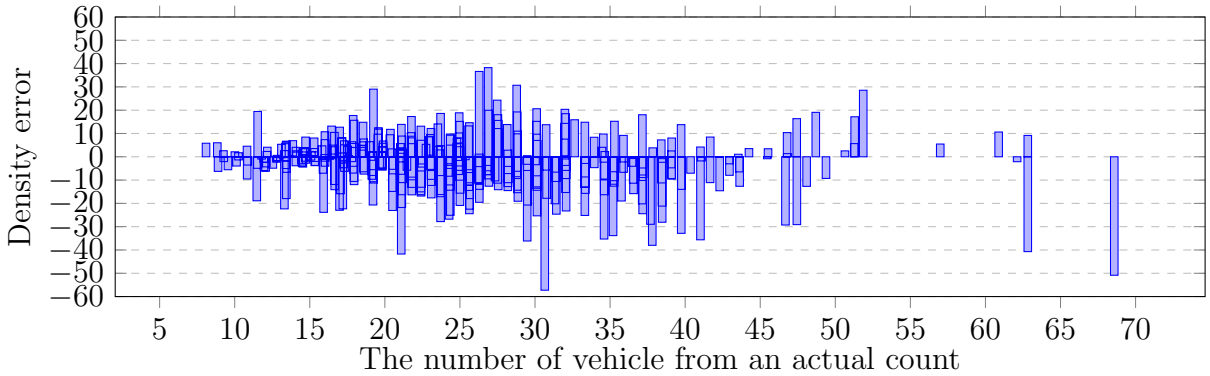
This subsection analyzes the empirical results for comparing related works to our proposed methods. Our prediction model was compared with the multi-column network for the crowd counting model consisting of MCNN [95], Hydra Net [61], and Tracount [78]. Besides, well-known classification networks with skip connections were analyzed (U-net [66], Resnet-101 [28], and Densenet-201 [33]) which have a large model structure. Their counting accuracy is shown in Table 5.3. The empirical result shows that our proposed model overcomes prediction models for crowd counting (MCNN, Hydra Net, and Tracount) in terms of MAE because their networks are more suitable for people counting. Meanwhile, well-known methods with skip connections have better counting accuracy, especially in Resnet-101 (93.33%). The proposed model with the highest accuracy (PT with dilated Conv) has similar accuracy (93.37%) to related works while requiring less trainable parameters and computational cost. Since traffic videos with adverse conditions are also included, the results from the proposed models are acceptable for their requirements (90%).

In term of time consumption (training and testing time), related models in Table 5.3 spend time consumption more than modified models in Tables 5.1 and 5.2. It proves that our proposed prediction model is overcome in terms of time consumption and model compactness. However, the problems related to high vehicle density and adverse conditions can not be solved by our methods.

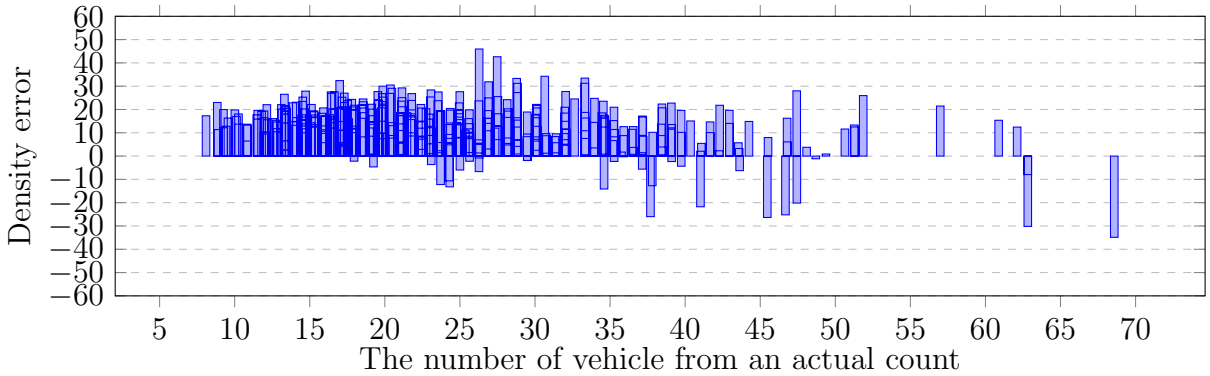
Figures 5.10 and 5.11 depict the density error generated by related works. In crowd counting models, the empirical results show that MCNN has high errors in both under and over-counting errors as shown in Figure 5.10a. On the other hand, Hydra Net and Tracount has a problem in only over-counting error as shown in Figures 5.10b and 5.10c, respectively. With a large model parameter, over and under-counting errors from model with skip connection has better results as shown in Figures 5.11a, 5.11b, and 5.11c respectively, where the difference between actual and predicted count does not exceed 40 vehicles.

| Model name | No. Trainable parameter | MAE | VA (%) | Training time (s) | Testing time (s) |
|---|---|---|---|---|---|
| B-2 (PT) with dilated Conv | 491,937 | 3.01 | 92.40 | 0.34 | 0.11 |
| MCNN [95] | 1,333,705 | 11.05 | 72.07 | 0.11 | 0.05 |
| Hydra Net [61] | 2,725,760 | 10.99 | 72.23 | 0.21 | 0.15 |
| Tracount [78] | 3,938,224 | 8.12 | 79.48 | 0.31 | 0.25 |
| U-net [66] | 24,339,281 | 5.13 | 84.50 | 0.48 | 0.18 |
| Resnet-101 [28] | 51,505,684 | 2.64 | 93.33 | 0.65 | 0.23 |
| Densenet-201 [33] | 26,933,105 | 3.6 | 90.09 | 0.78 | 0.26 |

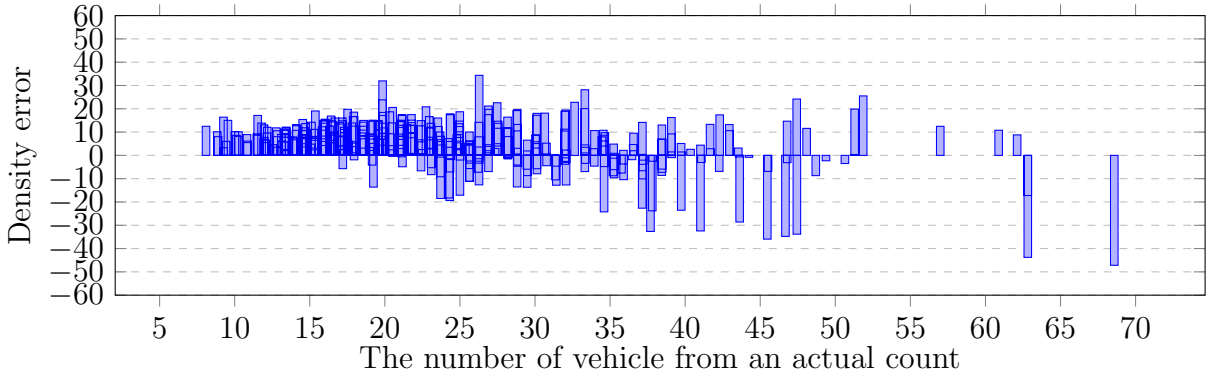
Table 5.3: Vehicle counting accuracy by using crowd counting model consisting of MCNN [95], Hydra Net [61], and Tracount [78] and well-known classification networks with skip connections consisting of U-net [66], Resnet-101 [28] , and Densenet-201 [33].




(a)

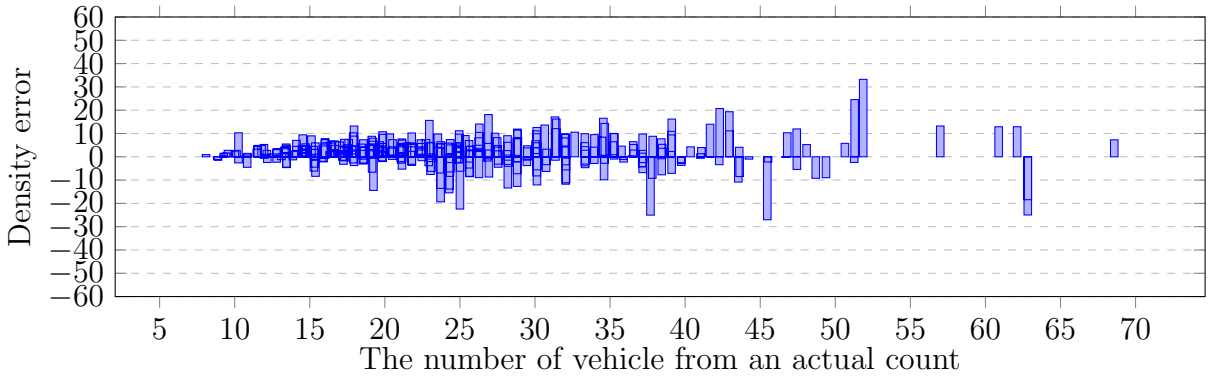



(b)

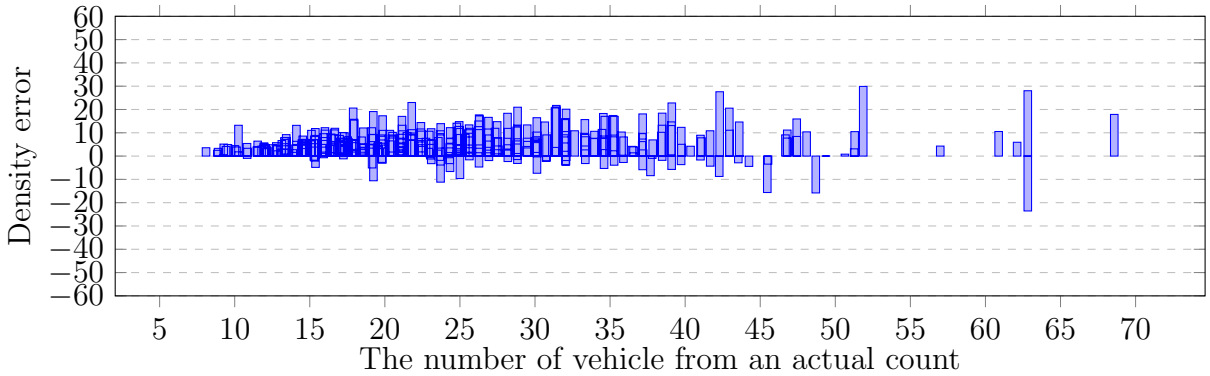



(c)

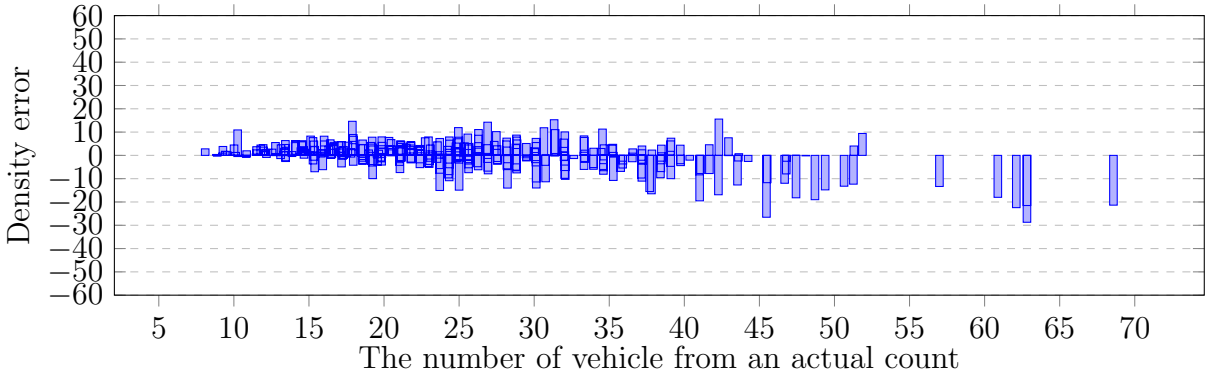
Figure 5.10: The density error from TRANCOS dataset [64] by using (a) MCNN [95], (b) Tra-count [78], and (c) Hydra-net [61].




(a)




(b)




(c)

Figure 5.11: The density error from TRANCOS dataset [64] by using (a) Resnet-101, (b) Densenet-201, and (c) U-net.

## 5.3 Analysis on the predicted density map

Since the density map was utilized for optimizing a prediction model and calculating the number of vehicles as a predicted count, the quality of a density map is one of the important factors for counting accuracy. As far as I am aware, there is no standard rule to measure density map quality. In this research, the predicted density map was evaluated by comparing its actual density map and classifying the density error which is either over-counting or under-counting error. The region of a predicted density map that has a higher or lower intensity than the actual density map is recognized as over-counting or under-counting error, respectively. Our predicted density map was observed in different aspects described as follows:

### 5.3.1 Object scale and vehicle density

In this research, low and high-level of features are considered as features of vehicles with small and large scales that are located in high and low vehicle density, respectively. Figures 5.12 and 5.13 show examples of traffic images with low (9 vehicles) and high (50 vehicles) vehicle density, respectively. Their input images are shown in (Figures 5.12a, 5.13a) and (Figures 5.12b, 5.13b) respectively. By improving low-level features with backward connections, the predicted density map by PT with dilated Conv is effective to formulate low vehicle density as shown in Figure 5.12c. Similar results are also formulated by other related works (e.g. MCNN (Figure 5.12d), Resnet-101 (Figure 5.12g), and U-net (Figure 5.12i)). Since the size of the object is related to the vehicle density and the camera viewpoint, prediction models can be misclassified between low and high density causing over-counting error as shown in Figures 5.12e, 5.12f, and 5.13h.

On the other hand, the traffic image with high vehicle density usually has various vehicle sizes, i.e. smaller and larger vehicles located at top and bottom of images as shown in Figure 5.13a. The experiment shows that crowd counting models, including the proposed method, is ineffective for counting small vehicle causing under-counting errors as shown in Figures 5.13c, 5.13d, 5.13e, and 5.13f. In addition, over-counting errors from misclassifciaiton still be occurred in their results. On the other hand, predicted density maps from Resnet-101 (Figure 5.13g), Densenet-201 (Figure 5.13h), and U-net (Figure 5.13i) has less sensitive to small vehicle sizes, where U-net has better counting accuracy in high vehicle density as shown in Figure 5.11c.

### 5.3.2 Traffic image quality

Since traffic images from the TRANCOS dataset are captured in different locations, their image quality is varied from outdoor environments and camera viewpoints. In the outdoor environment, the adverse condition (e.g. foggy, rainy, snowy, and so on) is a serious problem in this research because vehicles can not be seen clearly as shown in Figure 5.14a and its actual density map (Figure 5.14b). The empirical results show that predicted density maps from the proposed method can not be formulated in the area occluded by fog as shown in Figure 5.14c. Similar results also occurred in other crowded counting models (Figures 5.14d, 5.14e, and 5.14f) and semantic segmentation models (Figures 5.14g, 5.14h, and 5.14i). Therefore, the adverse condition is a major concern for an under-counting error.

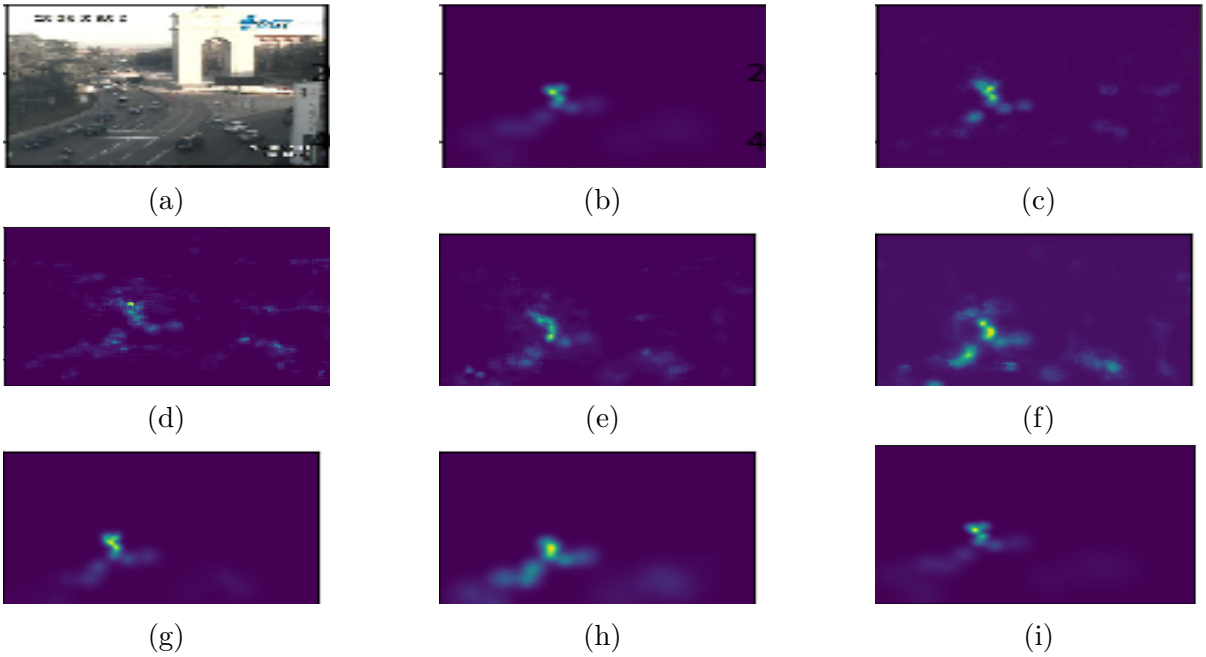



Figure 5.12: The example of prediction errors on the traffic image with low vehicle density (10 vehicles) consisting of (a) input image, (b) actual, and (c) predicted density maps by PT with dilated convolutions, (d) MCNN, (e) Hydra net, (f) Tracount, (g) Resnet-101, (h) Densenet-201, and (i) U-net.

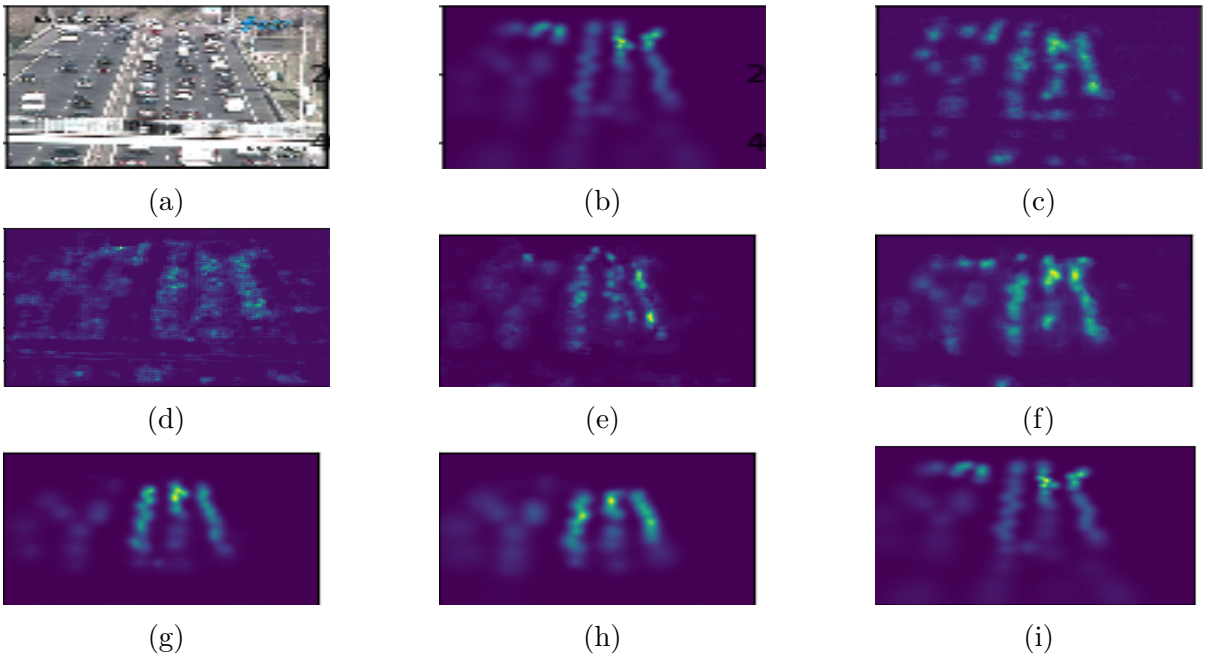



Figure 5.13: The example of prediction errors on the traffic image with high vehicle density (40 vehicles) consisting of (a) input image, (b) actual, and (c) predicted density maps by PT with dilated convolutions, (d) MCNN, (e) Hydra net, (f) Tracount, (g) Resnet-101, (h) Densenet-201, and (i) U-net.

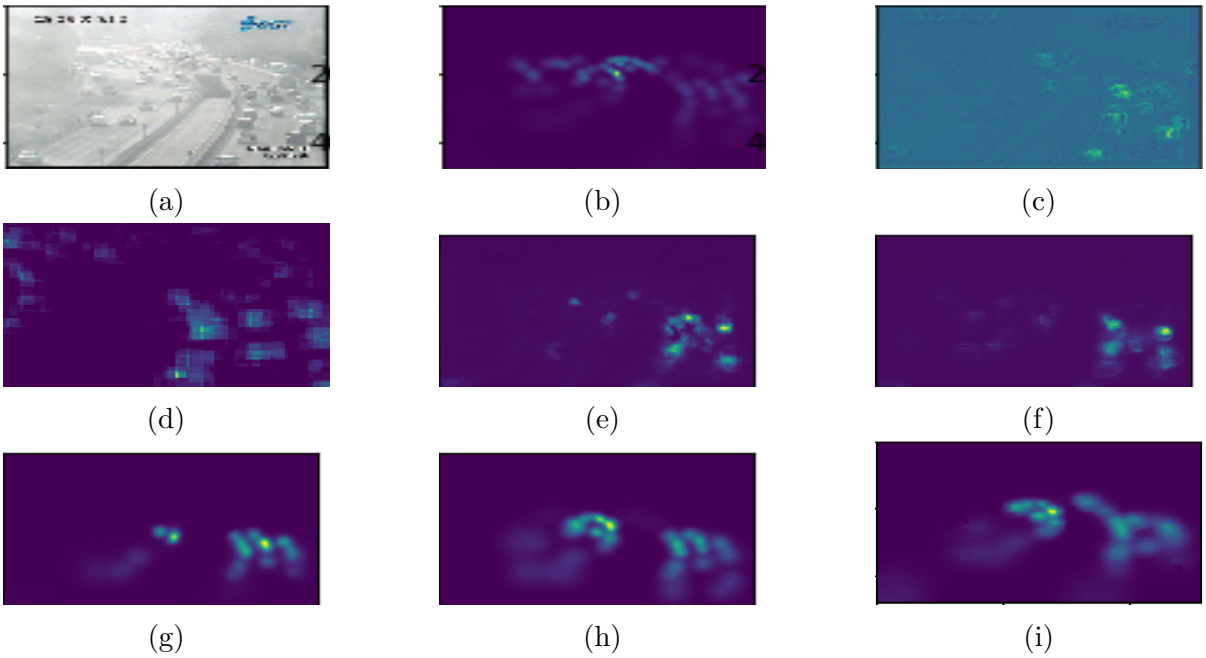



Figure 5.14: The example of prediction errors on the traffic image in an adverse condition consisting of (a) input image, (b) actual, and (c) predicted density maps by PT with dilated convolutions, (d) MCNN, (e) Hydra net, (f) Tracount, (g) Resnet-101, (h) Densenet-201, and (i) U-net.

On the other hand, Figure 5.15a shows an example of a traffic image with high brightness. Even though the ground truth (Figure 5.15b) indicated that all road regions should be formulated in the predicted density map, some part of them is still ignored in the prediction models. The experimental results show that the proposed method is still ineffective in regions containing small vehicles as shown in Figure 5.15c causing under-counting errors, but more regions were formulated than other related works, MCNN (Figure 5.15d), Hydra net (Figure 5.15e), Tracount (Figure 5.15f), and U-net (Figure 5.15i). With the large number of model parameters, the predicted density maps formulated by Resnet-101 and Densenet-201 is insensitive to this issue as shown in Figures 5.15g and 5.15h, respectively.

### 5.3.3 Traffic images with Non-vehicle objects

Besides under-counting errors caused by various vehicle density and image quality in previous subsections, the over-counting error is also concerned in this research. The major cause of the over-counting error is non-vehicle objects (e.g. traffic signs, bridges, buildings, and so on) which were formulated as vehicles in predicted density maps. This problem increases the number of vehicles in the traffic image. Figure 5.16 shows the example of traffic image in urban with several buildings where an input image and an actual density map are shown in Figures 5.16a and 5.16b, respectively. The experimental result shows that these counting errors usually be located outside the road regions as shown in the proposed predicted density map (Figure 5.16c) within red circles. The predicted density map of MCNN shown in Figure 5.16d has noise and distortion because of the combination among three CNNs. Other related crowd models, Hydra net (Figure

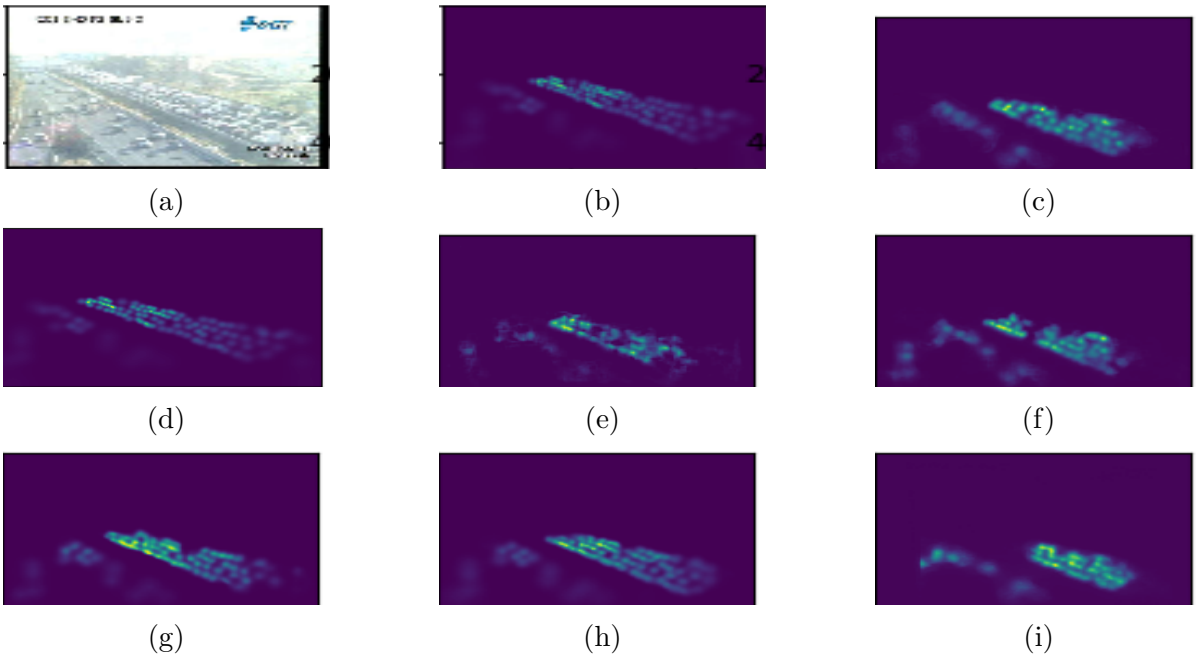



Figure 5.15: The example of prediction errors on the traffic image with high brightness consisting of (a) input image, (b) actual, and (c) predicted density maps by PT with dilated convolutions, (d) MCNN, (e) Hydra net, (f) Tracount, (g) Resnet-101, (h) Densenet-201, and (i) U-net.

5.16e) and Tracount (Figure 5.16f) have similar predicted density maps. These errors are caused by the variation of vehicle characteristics which might be overlapped with other non-vehicle objects. On the other hand, the remaining models can solve this significant problem as shown in Figures 5.16g, 5.16h, and 5.16i.

### 5.3.4 Effect on the dilated convolution

With the advantage of a dilated convolution, it is expected to preserve the semantic information in a predicted density map. In the previous section, Table 5.2 shows that PT with dilated convolutions has better counting accuracy. To prove our hypothesis in the predicted density map, two prediction models (PT with and without dilated Conv) were evaluated. These figures consist of input images (Figure 5.17a) and their predicted density maps produced by PT without dilated convolutions (Figures 5.17b) and PT with dilated convolutions (Figures 5.17c). For vehicle counting, empirical results show that dilated convolution generates more detail in predicted density maps, especially in high crowd density with small vehicles within red rectangular boxes. Therefore, it indicates that the error from the under-counting error is reduced in the proposed method.

## 5.4 Other applications

Since this prediction model is classified into a supervised learning technique for object counting, crowd counting (besides vehicle counting) can be tested in this experiment. This section describes the counting accuracy of our prediction model in other applications. In this experiment, there are two datasets for counting evaluation as follows:

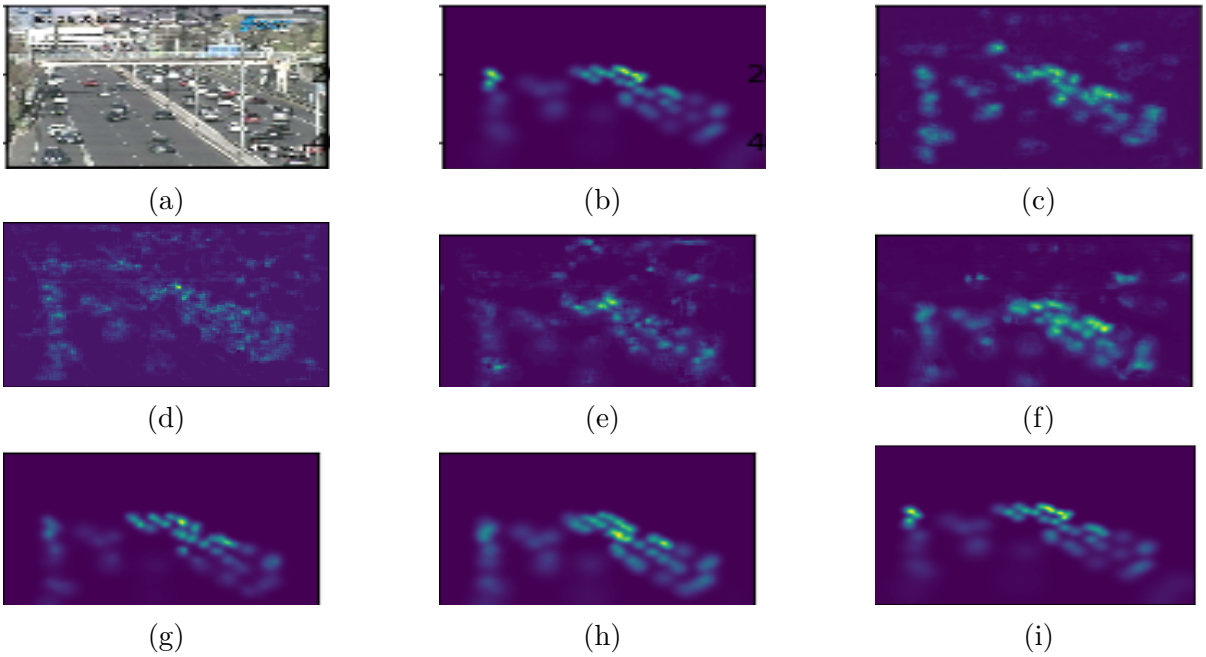



Figure 5.16: The example of prediction errors on the traffic image with non-vehicle objects consisting of (a) input image, (b) actual, and (c) predicted density maps by PT with dilated convolutions, (d) MCNN, (e) Hydra net, (f) Tracount, (g) Resnet-101, (h) Densenet-201, and (i) U-net.

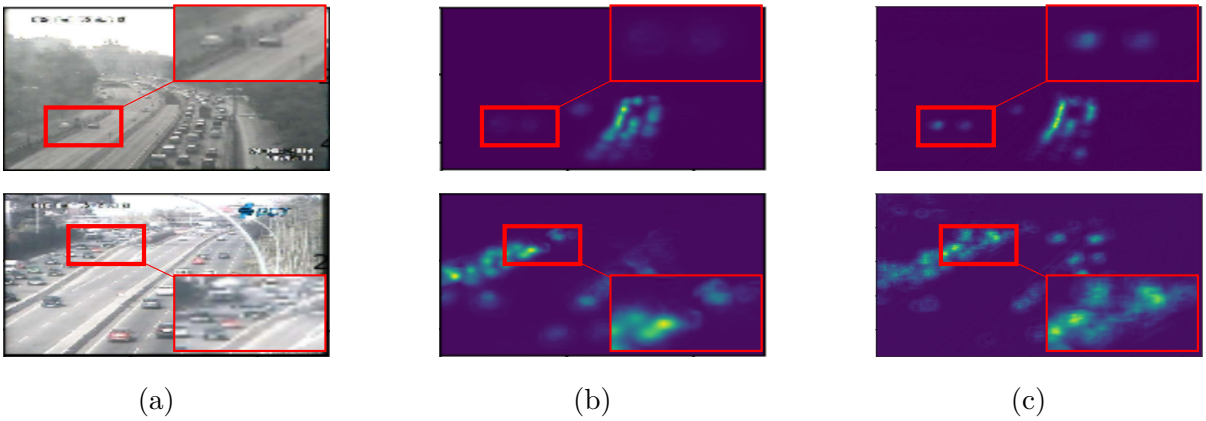



Figure 5.17: The example of effect on dilated convolutions from vehicle counting where (**a**) input images and their predicted density maps generated by (b) PT without and (c) with dilated convolutions, respectively.

Table 5.4: Counting accuracy in ShanghaiTech and UC_CC_50

| Prediction models | **ShanghaiTech A** | | **ShanghaiTech B** | | **UCF_CC_50** | |
|---|---|---|---|---|---|---|
| | RMSE | MAE | RMSE | MAE | RMSE | MAE |
| MCNN [95] | 173.2 | 110.2 | 41.2 | 26.2 | 509.1 | 377.6 |
| CP-CNN [74] | 106.2 | 73.6 | 30.1 | 20.1 | - | - |
| Switching-CNN [70] | 135 | 90.4 | 33.4 | 21.6 | - | - |
| HSRNet [96] | **100.3** | **62.3** | **11.8** | **7.2** | - | - |
| Deep-stacked [35] | 150.6 | 93.9 | 33.9 | 18.7 | - | - |
| C-CNN [72] | 141.7 | 88.7 | 22.1 | 14.9 | - | - |
| Rodriguez et al. [65] | - | - | - | - | **487.1** | 493.4 |
| Lempitsky et al. [43] | - | - | - | - | 541.6 | 419.5 |
| PT with dilated Conv | 165.56 | 87.31 | 19.41 | 15.76 | 507.96 | **358.83** |

- Shanghaitech Dataset: It is the dataset for crowd counting, targeting humans in the crowd image. This large-scale crowd counting dataset was introduced by [95]. It contains 1198 annotated images, with a total of $330,165$ people and the central coordinates of their heads. This dataset consists of

  - part A: There are 300 training and 182 testing images collected randomly crawled from the internet.

  - Part B: There are 400 training and 316 testing images taken from crowded scenes on Shanghai streets.

- UCF_CC_50 Dataset: Their crowd images were published by [36]. This dataset consists of 50 crowd images obtained from the internet. It is a very challenging dataset because The number of images is limited and the crowd count of the image changes dramatically. The total number of head (annotations) is $63,974$ for these 50 crowd images. The number of people is between 94 and $4,543$ with an average of $1,280$ individuals per image. Since this dataset provides a small number of crowd images, this dataset was evaluated by using estimation networks trained by the ShanghaiTech dataset (Part A and Part B).

The prediction model was evaluated by two evaluation metrics, MSE and MAE, which were calculated in Eqs. (3.8) and (3.9), respectively. the proposed method was evaluated and compared with other related works. The experimental results were obtained from their configurations. Table 5.4 illustrates the counting accuracy of the proposed method and other estimation networks. The counting accuracy in each dataset are described as follows:

In the first dataset, the experimental process has analyzed this dataset into two groups (Part A and Part B). As shown in Table 5.4, the counting accuracy from estimation networks in Part A is worse than that in Part B because the crowd images obtained from part A have a wide range of crowd density (from 0 to 1500 people) and image resolution. Therefore, the estimation network trained by a crowd image in part B can achieve more accurate results. Compared with counting accuracy, our empirical results cannot overcome other prediction models, especially CP-CNN [74] and HSRNet [96]. Figures 5.18 and 5.19 show density error from ShanghaiTech Part A and B, respectively, where HSRNet was utilized for comparison. In part A, figure 5.18a clearly shows that HSRNet trends have

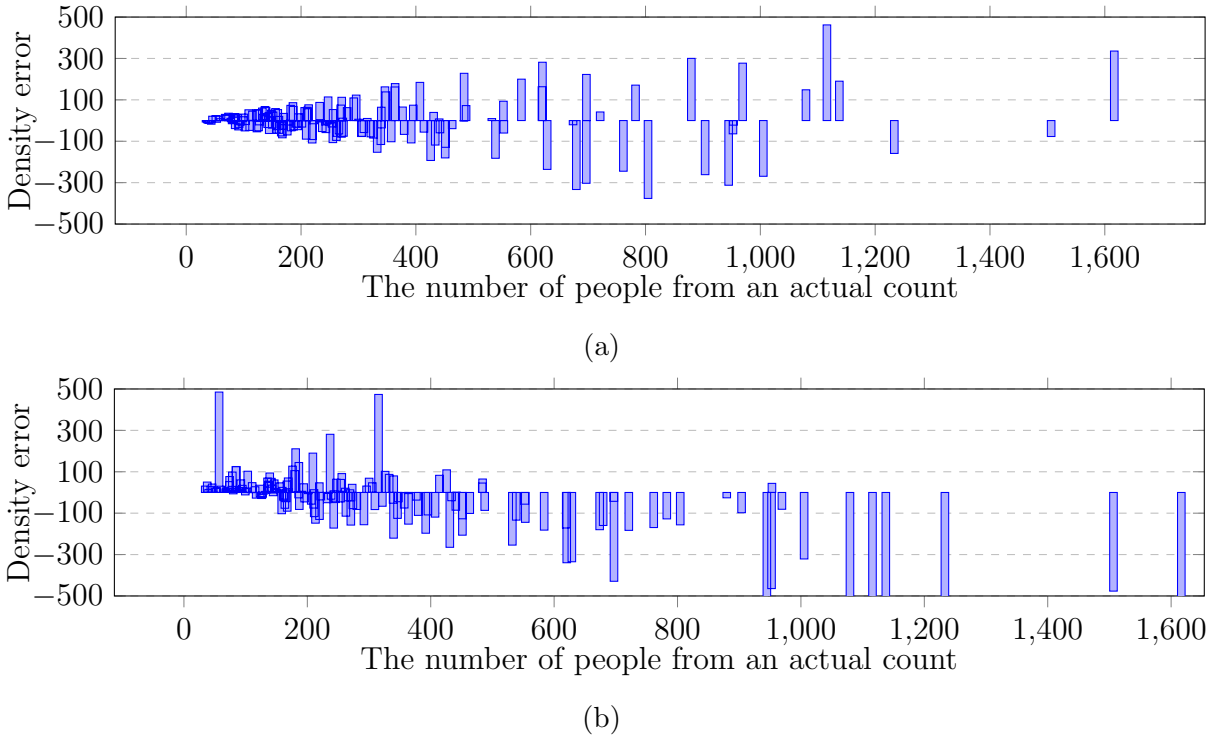



Figure 5.18: The density error from ShanghaiTech Part A by using (a) HSRNet [96] and (b) PT with dilated Conv.

over-counting errors. On the other hand, PT with dilated Conv has similar density errors from 0 to 400 people but it has a problem related to under-counting errors, especially in high crowd density (> 500 people). In part B, both counting accuracy from HSRNet (Figure 5.19a) and our models (Figure 5.19b) have less density error than part A because their image conditions (e.g. camera viewpoints, resolution, and so on) is similar. Similar to Part A, our counting accuracy in part B has an under-counting error in high crowd density as shown in Figure 5.19b.

In the second dataset, UCF_CC_50 contains a small number of crowd images. This experiment utilized pre-trained weights from ShanghaiTech Part A for evaluating the counting accuracy in UCF_CC_50 because both crowd images in UCF_CC_50 and part A are obtained from the different websites, where they have various resolutions and lighting conditions to handle the testing data (UCF_CC_50). Table 5.4 shows that our performance outperforms other estimation networks in MAE, but obtain lower robustness in RMSE. Compared with density error, both counting accuracy from Rodriguez et al. [65] (Figure 5.20a) and PT with dilated Conv (Figure 5.20b) have similar performance. However, they cannot handle a large number of people (> 1500 people), which is out of range in crowd images obtained from the ShanghaiTech Part A. From this experiment on human counting, it shows that our prediction models are not effective on ShanghaiTech and UCF_CC_50 datasets. based on the difference between vehicle and human counting, it can be described in the following aspects:

- The crowd density: Since humans have smaller sizes than vehicles in a crowded image, the crowded image from human counting usually has higher overlapping regions in its actual density map. It requires prediction models with a large number

of model parameters to improve the predicted density map quality and counting accuracy. Therefore, our compact model is less effective, especially in high crowd density.

- Camera viewpoints: The traffic surveillance cameras are typically located in specific locations. Then, vehicles are captured with a similar angle and camera viewpoints. However, the crowded images for human counting have various camera viewpoints. Therefore, the prediction model is hard to predict the number of the object under various image conditions.

## 5.5 Summary

This chapter describes the experiment for prediction models which were implemented with the same configuration for a fair comparison. Traffic images from different datasets were analyzed to select the most suitable dataset for evaluation. TRANCOS dataset was selected because there are more realistic traffic images that have low resolution and they are captured at the top views. The experiment is evaluated by vehicle density accuracy based on the Departments Approved Product List (APL). The empirical results show that the proposed models with skip connections have similar accuracy performance with other related works deploying less trainable parameters and time consumption. On the other hand, dilated convolution can help to increase counting accuracy and recover regions with small objects in the predicted density map. However, it reveals that the factor related to time complexity is not only the number of parameters but The backward connections (MS, BT, and PT) are other factors for computation time. In addition, there are significant errors where the prediction models are ineffective in images with high crowd density and adverse conditions, as shown in an experiment on human counting.

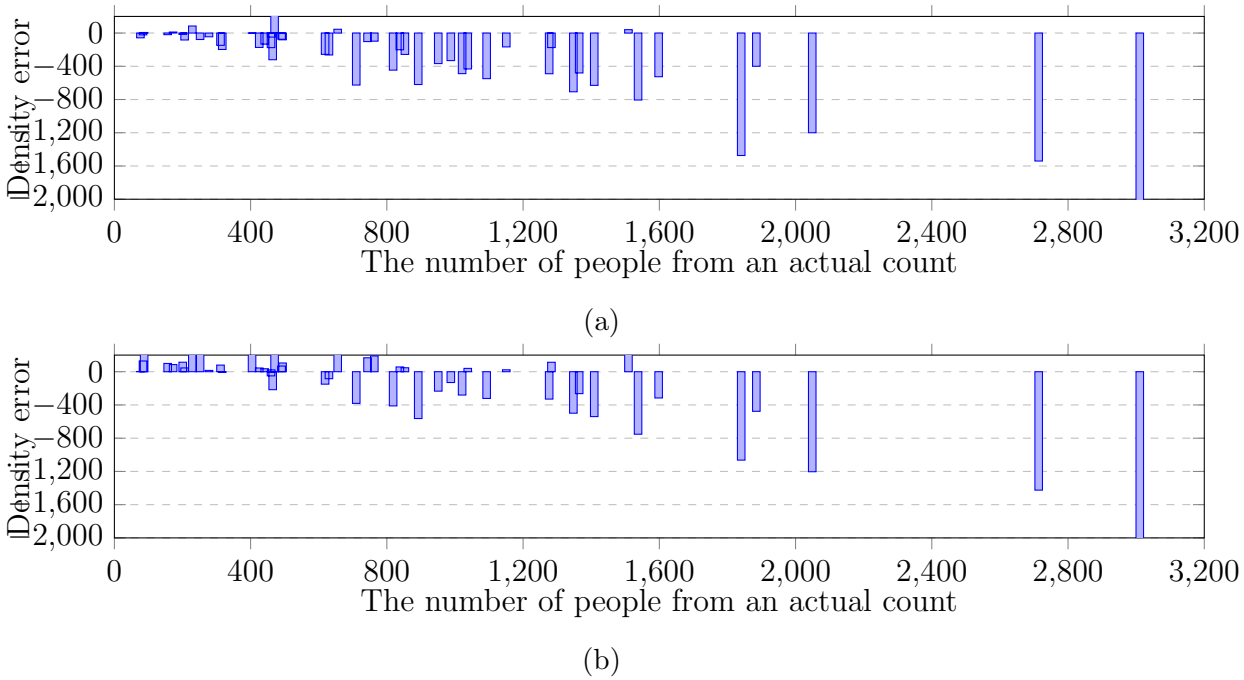



Figure 5.20: The density error from UCF_CC_50 by using (a) Rodriguez et al. [65] and (b) PT with dilated Conv.

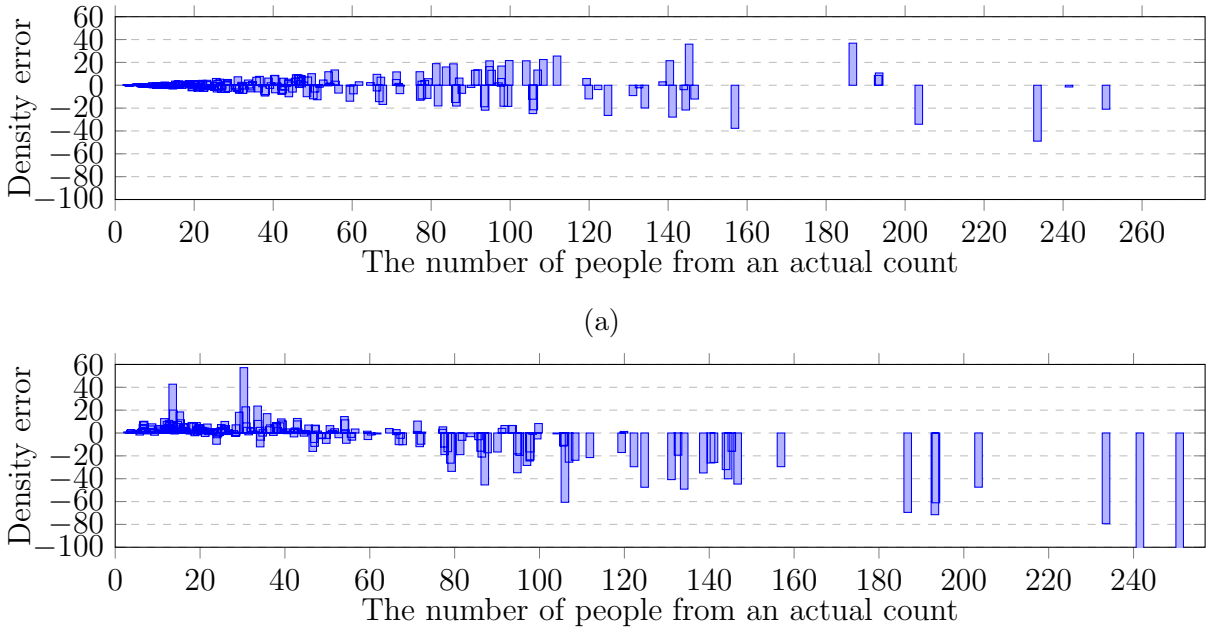



Figure 5.19: The density error from ShanghaiTech Part B by using (a) HSRNet [96] and (b) PT with dilated Conv.

# Chapter 6

# Conclusion and Future work

## 6.1 Conclusion

The surveillance system is widely used for analyzing the characteristic of target density (e.g. human, vehicle, and so on) in specific areas and formulate the number of targets. Traffic light control system (TLCS) is one of the services of Vehicle Imaging Vehicle detection system (VIVDS) to control the optimal time for green, yellow, red lights, after acquiring the vehicle density from sensors. Nowadays, there are several sensors for counting or estimating the number of vehicles on the road (e.g. loop detector, ultrasonic camera, surveillance system, and so on), where surveillance system is widely deployed because they provide low cost of installation and maintenance. Since the surveillance system is operated in the outdoor environment with a low specification device installed in RSU, the density estimation should be insensitive to lighting conditions and has less time consumption. Even though recent estimation methods using deep learning-based techniques achieved much higher accuracy, they require a large number of model parameters for training. Therefore this study is focused on the single-column model which has fewer parameters and computational cost. Besides, the proposed method should handle various traffic video conditions (e.g. lighting conditions, and so on).

It is well-known that density estimation or object counting is categorized into the detection and regression-based approaches. The detection-based approach is initially introduced for estimation, where target regions are extracted and counted. A recent study related to detection-based approaches has two types (appearance and motion-based methods). The appearance-based approach analyzes the targets texture and characteristics for extracting their regions. However, their appearances are sensitive to video conditions (e.g. camera viewpoint, lighting conditions, and so on). On the other hand, motion-based approaches are detection techniques without pre-knowledge information which is insensitive in any environment, but video sequences are required for this method. Then, the deep learning-based technique is introduced with promising performance. The most popular detection-based approach using deep learning is RCNN utilizing region proposal and CNN. However, this method requires a large number of model parameters.

Since a detection-based approach has a limitation on the occlusion problem, it is ineffective to an image with a large number of targets obtained from a surveillance system. This approach extracts holistic features from a whole input image instead of extracting a specific feature from each target. Regression-based approaches establish a direct mapping between these properties and the number of targets in the image called density map

while avoiding actual segmentation of individual or track of features. The main challenge is focused on scale-aware to handle various vehicles within images. Recent studies also utilize multiple CNNs to handle objects of different sizes, but more network parameters are required. Therefore, our main research is concerned with a single-column method requiring only one CNN to minimize computational cost.

This dissertation contributes to density-aware traffic density estimation from traffic images in different traffic video conditions. The proposed method formulates the density map using average distances of the vehicle instead of vehicle sizes because the actual sizes cannot be accurately obtained in practical circumstances. Therefore, our density maps were computed by convoluting with a Gaussian kernel in different spread parameters for different locations. These techniques enable to visualize the different patterns between the region with low and high vehicle density, more than utilizing traditional methods in different camera viewpoints.

The proposed prediction model was designed from one stack of MCNN. They are modified by deploying a convolutional layer with the smallest kernel sizes while preserving the receptive field sizes. The main contribution of this study is to utilize skip connections to improve the accuracy of the single-column model. The skip connections were designed as forward and backward connections that extract low-level and high-level feature maps to emphasize deep and shallow layers, respectively. Since CNN typically is designed as a feed-forward model, the high-level feature maps can not be extracted directly. This dissertation proposes three techniques to handle this problem. First, another model is designed as slave networks to only generate high-level features. Second, the high-level features can be transferred from a trained prediction model in different images called batch transfer. Third, the slave network can be used as a pre-trained network for the master network, where this technique is called a pre-trained map. In addition, pooling and up-sampling layers were replaced with dilated convolutions to avoid information loss while resizing.

All experiments were run by Python with Tensorflow to fairly compare accuracy performance and computational cost. The dataset is obtained from the TRANCOS dataset because their traffic images are more realistic to the practical application. Besides, human counting is evaluated in this experiment, where their images were obtained from Shanghaitech and UCF_CC_50 datasets. The experimental results show that all hypotheses described in our contribution are achieved as follows:

- The proposed density map using estimated vehicle size by average distance overcome the traditional method by providing more information to distinguish regions with high from low density.
- The proposed prediction model achieved comparable accuracy with related works (especially Resnet-101) while reducing the number of model parameters. In addition, both of them fulfills the requirement in VIVDS.

## 6.2 Future work

Since this research has been studied for a half-century, there have always some aspects of any work that can be improved and expanded upon by future research. This may be caused by the limitations of time and resources, or other unforeseen difficulties. Our

research works might remain an unsolved problem and new topics appear continuously. Therefore, it is the wish of the author to suggest some ideas regarding the future direction of this work which is summarized as follow:

- A routing algorithm for skip connections will be explored to handle an object with a small scale and high density, especially in human counting.
- Since our algorithm is simulated in hardware with high performance, the practical circumstances will be implemented by using hardware used in real applications. Then, the minimum requirement related to time consumption can be decided.
- The factors related to time consumption will be analyzed. The experiment indicated that our models with skip connection have much amount of training and testing time even though their number of parameters is the same. It means that there are other factors related to computational cost where they can be further analyzed to solve this problem.
- In the proposed method, sizes and scales of vehicles are formulated by using their spatial information (average distances) to compute density maps, where the vehicle with large and small scale should be located at low and high traffic density, respectively. The experimental results show that this hypothesis might fail on traffic images with only high traffic density and small scales. The computation density maps should be fixed and they should represent the objects with different scales.

# Bibliography


[1] Vahid Abolghasemi and Alireza Ahmadyfard. "An edge-based color-aided method for license plate detection". In: *Image and Vision Computing* 27.8 (2009), pp. 1134–1142.

[2] Sikandar Amin and Fabio Galasso. "Geometric proposals for faster r-cnn". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. (2017), pp. 1–6.

[3] Dmitriy Anisimov and Tatiana Khanova. "Towards lightweight convolutional neural networks for object detection". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. (2017), pp. 1–8.

[4] Jon Arróspide, Luis Salgado, and Massimo Camplani. "Image-based on-road vehicle detection using cost-effective histograms of oriented gradients". In: *Journal of Visual Communication and Image Representation* 24.7 (2013), pp. 1182–1190.

[5] H Asaidi, A Aarab, and M Bellouki. "Shadow elimination and vehicles classification approaches in traffic video surveillance context". In: *Journal of Visual Languages & Computing* 25.4 (2014), pp. 333–345.

[6] National Electrical Manufacturers Association et al. "NEMA Standards Publication TS 2-2003 v02. 06–Traffic Controller Assemblies with NTCIP Requirements". In: *Rosslyn, Virginia, USA* (2003).

[7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. "Surf: Speeded up robust features". In: *European conference on computer vision*. Springer. (2006), pp. 404–417.

[8] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[9] Lokesh Boominathan, Srinivas SS Kruthiventi, and R Venkatesh Babu. "Crowdnet: A deep convolutional network for dense crowd counting". In: *Proceedings of the 24th ACM international conference on Multimedia*. ACM. (2016), pp. 640–644.

[10] Iyadh Cabani, Gwenaelle Toulminet, and Abdelaziz Bensrhair. "Color-based detection of vehicle lights". In: *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.* IEEE. (2005), pp. 278–283.

[11] Antoni B Chan, Zhang-Sheng John Liang, and Nuno Vasconcelos. "Privacy preserving crowd monitoring: Counting people without people models or tracking". In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. (2008), pp. 1–7.

[12] Antoni B Chan and Nuno Vasconcelos. "Modeling, clustering, and segmenting video with mixtures of dynamic textures". In: *IEEE transactions on pattern analysis and machine intelligence* 30.5 (2008), pp. 909–926.

[13] Yi-Ming Chan et al. "Vehicle detection under various lighting conditions by incorporating particle filter". In: *2007 IEEE Intelligent Transportation Systems Conference*. IEEE. (2007), pp. 534–539.

[14] Duan-Yu Chen, Yu-Hao Lin, and Yang-Jie Peng. "Nighttime brake-light detection by Nakagami imaging". In: *IEEE Transactions on Intelligent Transportation Systems* 13.4 (2012), pp. 1627–1637.

[15] Duan-Yu Chen and Yang-Jie Peng. "Frequency-tuned taillight-based nighttime vehicle braking warning system". In: *IEEE Sensors Journal* 12.11 (2012), pp. 3285–3292.

[16] Minkyu Cheon et al. "Vision-based vehicle detection system with consideration of the detecting location". In: *IEEE transactions on intelligent transportation systems* 13.3 (2012), pp. 1243–1252.

[17] Antonio Criminisi, Jamie Shotton, Ender Konukoglu, et al. "Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning". In: *Foundations and Trends® in Computer Graphics and Vision* 7.2–3 (2012), pp. 81–227.

[18] Jifeng Dai et al. "Deformable convolutional networks". In: *Proceedings of the IEEE international conference on computer vision*. (2017), pp. 764–773.

[19] Vincent Dumoulin and Francesco Visin. "A guide to convolution arithmetic for deep learning". In: *arXiv preprint arXiv:1603.07285* (2016).

[20] Ahmed Elgammal, David Harwood, and Larry Davis. "Non-parametric model for background subtraction". In: *European conference on computer vision*. Springer. (2000), pp. 751–767.

[21] Arlington County Department of Environmental Services. *Arlington County Traffic Signal Specifications*. `https://transportation.arlingtonva.us/wp-content/uploads/sites/19/2017/09/Arlington-County-Traffic-Signal-Specifications.pdf`. [Online; accessed 30-Sep-2017]. 2017.

[22] M. Everingham et al. *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[23] Rogerio Schmidt Feris et al. "Large-scale vehicle detection, indexing, and search in urban surveillance videos". In: *IEEE Transactions on Multimedia* 14.1 (2011), pp. 28–42.

[24] Andreas Geiger et al. "Vision meets Robotics: The KITTI Dataset". In: ().

[25] Paul Geladi and Bruce R Kowalski. "Partial least-squares regression: a tutorial". In: *Analytica chimica acta* 185 (1986), pp. 1–17.

[26] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2014), pp. 580–587.

[27] Robert M Haralick, Karthikeyan Shanmugam, et al. "Textural features for image classification". In: *IEEE Transactions on systems, man, and cybernetics* 6 (1973), pp. 610–621.

[28] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[29] Byeongho Heo, Kimin Yun, and Jin Young Choi. "Appearance and motion based deep learning architecture for moving object detection in moving camera". In: *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE. (2017), pp. 1827–1831.

[30] John Honovich. *Average Frame Rate Video Surveillance 2016*. `https://ipvm.com/reports/avg-frame-rate-2016`. [Online; accessed 11-Feb-2016]. 2016.

[31] Thanarat Horprasert, David Harwood, and Larry S Davis. "A statistical approach for real-time robust background subtraction and shadow detection". In: *Ieee iccv*. Vol. 99. 1999. Citeseer. (1999), pp. 1–19.

[32] PeiFeng Hu, Zong Tian, George Bebis, et al. *Evaluation of Video Detection Systems and Development of Application Guidelines at Signalized Intersections*. Tech. rep. Nevada. Dept. of Transportation, 2010.

[33] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[34] Jonathan Huang et al. "Speed/accuracy trade-offs for modern convolutional object detectors". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2017), pp. 7310–7311.

[35] Siyu Huang et al. "STACKED POOLING FOR BOOSTING SCALE INVARIANCE OF CROWD COUNTING". In: ().

[36] Haroon Idrees et al. "Multi-source multi-scale counting in extremely dense crowd images". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2013, pp. 2547–2554.

[37] Neeraj K Kanhere and Stanley T Birchfield. "Real-time incremental segmentation and tracking of vehicles at low camera angles using stable features". In: *IEEE transactions on intelligent transportation systems* 9.1 ((2008)), pp. 148–160.

[38] KP Karman. "Moving object segmentation based on adaptive reference images". In: *Proc. of EUSIPCO, Barcelona* (1990).

[39] Jeremy Karnowski. *Alexnet visualization*. `https://jeremykarnowski.wordpress.com/2015/07/15/alexnet-visualization/#more-143`. 2015.

[40] Nima Khairdoost, S Amirhassan Monadjemi, and Kamal Jamshidi. "Front and rear vehicle detection using hypothesis generation and verification". In: *Signal & Image Processing* 4.4 (2013), p. 31.

[41] Kyungnam Kim et al. "Real-time foreground–background segmentation using codebook model". In: *Real-time imaging* 11.3 (2005), pp. 172–185.

[42] Shohei Kumagai, Kazuhiro Hotta, and Takio Kurita. "Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting". In: *arXiv preprint arXiv:1703.09393* (2017).

[43] Victor Lempitsky and Andrew Zisserman. "Learning to count objects in images". In: *Advances in neural information processing systems*. 2010, pp. 1324–1332.

[44] Jing Li and Nigel M Allinson. "A comprehensive review of current local features for computer vision". In: *Neurocomputing* 71.10-12 (2008), pp. 1771–1787.

[45] Ye Li and Fei-Yue Wang. "Vehicle detection based on and–or graph and hybrid image templates for complex urban traffic conditions". In: *Transportation Research Part C: Emerging Technologies* 51 (2015), pp. 19–28.

[46] Yuhong Li, Xiaofan Zhang, and Deming Chen. "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 1091–1100.

[47] Todd Litman. "Congestion Costing Critique: Critical Evaluation of the Urban Mobility Report". In: (2013).

[48] Wei Liu et al. "Rear vehicle detection and tracking for lane change assist". In: *2007 IEEE intelligent vehicles symposium*. IEEE. (2007), pp. 252–257.

[49] David G Lowe et al. "Object recognition from local scale-invariant features." In: *iccv*. Vol. 99. 2. (1999), pp. 1150–1157.

[50] Chen Change Loy et al. "Crowd counting and profiling: Methodology and evaluation". In: *Modeling, simulation and visual analysis of crowds*. Springer, 2013, pp. 347–382.

[51] Zhiming Luo et al. "Traffic analytics with low-frame-rate videos". In: *IEEE Transactions on Circuits and Systems for Video Technology* 28.4 (2016), pp. 878–891.

[52] Vishal Maini. *Machine Learning for Humans, Part 4: Neural Networks  Deep Learning*. `https://medium.com/machine-learning-for-humans/neural-networks-deep-learning-cdad8aeae49b`. 2017.

[53] R Manikandan and R Ramakrishnan. "Video object extraction by using background subtraction techniques for sports applications". In: *Digital Image Processing* 5.9 (2013), pp. 435–440.

[54] Antoine Manzanera and Julien C Richefeu. "A new motion detection algorithm based on $\Sigma$–$\Delta$ background estimation". In: *Pattern Recognition Letters* 28.3 (2007), pp. 320–328.

[55] Aparecido Nilceu Marana et al. "Estimating crowd density with Minkowski fractal dimension". In: *1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258)*. Vol. 6. IEEE. (1999), pp. 3521–3524.

[56] Mark Marsden et al. "Fully convolutional crowd counting on highly congested scenes". In: *arXiv preprint arXiv:1612.00220* (2016).

[57] Kenan Mu et al. "Multiscale edge fusion for vehicle detection based on difference of Gaussian". In: *Optik* 127.11 (2016), pp. 4794–4798.

[58] Kapileswar Nellore and Gerhard Hancke. "A survey on urban traffic management system using wireless sensor networks". In: *Sensors* 16.2 (2016), p. 157.

[59] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 7 (2002), pp. 971–987.

[60] Ronan O'Malley, Edward Jones, and Martin Glavin. "Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions". In: *IEEE Transactions on Intelligent Transportation Systems* 11.2 (2010), pp. 453–462.

[61] Daniel Onoro-Rubio and Roberto J López-Sastre. "Towards perspective-free object counting with deep learning". In: *European Conference on Computer Vision*. Springer. (2016), pp. 615–629.

[62] Ronan OMalley, Martin Glavin, and Edward Jones. "Vehicle detection at night based on tail-light detection". In: *1st international symposium on vehicular computing systems, Trinity College Dublin*. (2008).

[63] Mritunjay Rai et al. "Advance Intelligent Video Surveillance System (AIVSS): A Future Aspect". In: *Intelligent Video Surveillance*. IntechOpen, 2018.

[64] Roberto Lpez-Sastre Saturnino Maldonado Bascn Ricardo Guerrero-Gmez-Olmedo Beatriz Torre-Jimnez and Daniel Ooro-Rubio. "Extremely Overlapping Vehicle Counting". In: *Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA)*. (2015).

[65] Mikel Rodriguez et al. "Density-aware person detection and tracking in crowds". In: *2011 International Conference on Computer Vision*. IEEE. 2011, pp. 2423–2430.

[66] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

[67] Wojciech Rosinski. *Deep Learning Frameworks Speed Comparison*. `https://wrosinski.github.io/deep-learning-frameworks/`. 2017.

[68] Sitapa Rujikietgumjorn and Nattachai Watcharapinchai. "Vehicle detection with sub-class training using R-CNN for the UA-DETRAC benchmark". In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE. (2017), pp. 1–5.

[69] CE Rusmassen and CKI Williams. *Gaussian process for machine learning*. 2005.

[70] Deepak Babu Sam, Shiv Surya, and R Venkatesh Babu. "Switching convolutional neural network for crowd counting". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE. (2017), pp. 4031–4039.

[71] John Shawe-Taylor, Nello Cristianini, et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[72] Xiaowen Shi et al. "A Real-Time Deep Network for Crowd Counting". In: *arXiv preprint arXiv:2002.06515* (2020).

[73] Vishwanath A Sindagi and Vishal M Patel. "A survey of recent advances in cnn-based single image crowd counting and density estimation". In: *Pattern Recognition Letters* 107 (2018), pp. 3–16.

[74] Vishwanath A Sindagi and Vishal M Patel. "Generating high-quality crowd density maps using contextual pyramid cnns". In: *Proceedings of the IEEE International Conference on Computer Vision*. 2017, pp. 1861–1870.

[75] Sayanan Sivaraman and Mohan Manubhai Trivedi. "A general active-learning framework for on-road vehicle recognition and tracking". In: *IEEE Transactions on Intelligent Transportation Systems* 11.2 (2010), pp. 267–276.

[76] Sayanan Sivaraman and Mohan Manubhai Trivedi. "Vehicle detection by independent parts for urban driver assistance". In: *IEEE Transactions on Intelligent Transportation Systems* 14.4 (2013), pp. 1597–1608.

[77] Chris Stauffer and W. Eric L. Grimson. "Learning patterns of activity using real-time tracking". In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 747–757.

[78] Shiv Surya. "TraCount: a deep convolutional neural network for highly overlapping vehicle counting". In: *Proceedings of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing*. 2016, pp. 1–6.

[79] Bin Tian et al. "Rear-view vehicle detection and tracking by combining multiple parts for complex urban surveillance". In: *IEEE Transactions on Intelligent Transportation Systems* 15.2 (2013), pp. 597–606.

[80] Luo-Wei Tsai, Jun-Wei Hsieh, and Kuo-Chin Fan. "Vehicle detection using normalized color and edge map". In: *IEEE transactions on Image Processing* 16.3 (2007), pp. 850–864.

[81] Vladimir Vapnik. *The nature of statistical learning theory*. Springer science & business media, 2013.

[82] Paul Viola, Michael Jones, et al. "Rapid object detection using a boosted cascade of simple features". In: *CVPR (1)* 1 (2001), pp. 511–518.

[83] Hai Wang, Chaochun Yuan, and Yingfeng Cai. "Smart road vehicle sensing system based on monocular vision". In: *Optik* 126.4 (2015), pp. 386–390.

[84] Luyang Wang et al. "Skip-connection convolutional neural network for still image crowd counting". In: *Applied Intelligence* 48.10 (2018), pp. 3360–3371.

[85] Shuo Wang, Koray Ozcan, and Anuj Sharma. "Region-based deformable fully convolutional networks for multi-class object detection at signalized traffic intersections: NVIDIA AICity challenge 2017 Track 1". In: *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*. IEEE. 2017, pp. 1–4.

[86] Xueming Wang et al. "Vision-based two-step brake detection method for vehicle collision avoidance". In: *Neurocomputing* 173 (2016), pp. 450–461.

[87] Yi Wang et al. "CDnet 2014: an expanded change detection benchmark dataset". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. (2014), pp. 387–394.

[88] Longyin Wen et al. "UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking". In: *arXiv preprint arXiv:1511.04136* (2015).

[89] Longyin Wen et al. "UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking". In: *arXiv CoRR* abs/1511.04136 (2015).

[90] Xuezhi Wen et al. "Efficient feature selection and classification for vehicle detection". In: *IEEE Transactions on Circuits and Systems for Video Technology* 25.3 (2014), pp. 508–517.

[91] Xinyu Wu et al. “Crowd density estimation using texture analysis and learning”. In: *2006 IEEE international conference on robotics and biomimetics*. IEEE. (2006), pp. 214–219.

[92] Lunhui Xu and Wenping Bu. “Traffic flow detection method based on fusion of frames differencing and background differencing”. In: *2011 Second International Conference on Mechanic Automation and Control Engineering*. IEEE. (2011), pp. 1847–1850.

[93] Zi Yang and Lilian SC Pun-Cheng. “Vehicle detection in intelligent transportation systems and its applications under varying environments: A review”. In: *Image and Vision Computing* 69 (2018), pp. 143–154.

[94] Akio Yoneyama, Chia-Hung Yeh, and C-C Jay Kuo. “Robust vehicle and traffic information extraction for highway surveillance”. In: *EURASIP Journal on Advances in Signal Processing* 2005.14 (2005), pp. 2305–2321.

[95] Yingying Zhang et al. “Single-image crowd counting via multi-column convolutional neural network”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. (2016), pp. 589–597.

[96] Zhikang Zou et al. “Crowd Counting via Hierarchical Scale Recalibration Network”. In: *arXiv preprint arXiv:2003.03545* (2020).

# Publication

## International journal

[1] Sorn Sooksatra, Toshiaki Kondo, Pished Bunnun, and Atsuo Yoshitaka. "Headlight recognition for night-time traffic surveillance using spatial-temporal information", *Singal, Image, and Video Processing*. 2020, 14(1), pp. 1-8.

[2] Sorn Sooksatra, Toshiaki Kondo, Pished Bunnun, and Atsuo Yoshitaka. "Redesigned Skip-Network for Crowd Counting with Dilated Convolution and Backward Connection", *Journal of Imaging*. 2020, 6(5), pp. 28-43.

## Domestic conference

[3] Sorn Sooksatra, Toshiaki Kondo, Pished Bunnun, and Atsuo Yoshitaka. "Headlights classification for traffic surveillance using a structure tensor method with PADI", $55^{th}$ *Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*. Tsukuba, Japan, 2016, pp. 1472-1477.

## International conference

[4] Sorn Sooksatra, Atsuo Yoshitaka, Toshiaki Kondo, and Pished Bunnun, "The Density-Aware Estimation Network for Vehicle Counting in Traffic Surveillance System", $15^{th}$ *International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, Sorrento, Italy, 2019, pp. 231-238.