A SCENE INVARIANT CONVOLUTIONAL NEURAL NETWORK FOR VISUAL CROWD COUNTING USING FAST-LANE AND SAMPLE SELECTIVE METHODS

TEOH SHEN KHANG

DOCTOR OF PHILOSOPHY (ENGINEERING)

FACULTY OF ENGINEERING AND GREEN
TECHNOLOGY
UNIVERSITI TUNKU ABDUL RAHMAN
JULY 2023

A SCENE INVARIANT CONVOLUTIONAL NEURAL NETWORK FOR VISUAL CROWD COUNTING USING FAST-LANE AND SAMPLE SELECTIVE METHODS

By

TEOH SHEN KHANG

A thesis submitted to the Department of Electronic Engineering,
Faculty of Engineering and Green Technology,
Universiti Tunku Abdul Rahman,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Engineering
July 2023

ABSTRACT

A SCENE INVARIANT CONVOLUTIONAL NEURAL NETWORK FOR VISUAL CROWD COUNTING USING FAST-LANE AND SAMPLE SELECTIVE METHODS

Teoh Shen Khang

Convolutional neural network (CNN) based crowd counting aims to estimate the number of pedestrians from the image. Existing research usually follow the training-testing protocol within a single dataset and the accuracy drops when conducting cross-dataset evaluation. Density map prediction methodology is widely used but it has drawbacks in ground truth generation and the use of Euclidean distance results in low quality density map. Additionally, CNN models face the challenges of vanishing gradients and zero weights, leading to low accuracy in predictions. This study uses global regression methodology and whole image-based training pattern to directly estimates the final count from image. The proposed model is designed with single column architecture using single filter size and max pooling size. Fast lane connection and sample selective algorithms have been designed specifically to tackle the issue of vanishing gradient and enhance the quality of the model. The performance of the proposed model, which is scene-invariant, was assessed using the ShanghaiTech dataset, the UCSD dataset, and the Mall dataset. It achieved an average MAE of 2.75 and a MSE of 3.65. As a result of the proposed method, the model performs well overall and exhibits improved generalisability to unseen scenes.

ACKNOWLEDGEMENTS

I am heartily thankful to my supervisors, Associate Professor Dr Yap Vooi Voon and Professor Humaira Nisar, whose encouragement, supervision, and support from the preliminary to the concluding levels enabled me to develop an understanding of the subject. Their guidance helped me in all the time of research and writing of this thesis. I would especially like to thank my family. My wife, Tan Lyk Yin has been extremely supportive of me throughout this entire process and has made countless sacrifices to help me get to this point. I would also extend my gratitude to UTAR for supporting my study through the UTAR staff scholarship. This research was supported by a Fundamental Research Grant Scheme (FRGS) funded from the Ministry of Higher Education (MOHE) Project No. FRGS/1/2017/ICT02/UTAR/03/1.

APPROVAL SHEET

This thesis entitled "**A SCENE INVARIANT CONVOLUTIONAL NEURAL NETWORK FOR VISUAL CROWD COUNTING USING FAST-LANE AND SAMPLE SELECTIVE METHODS**" was prepared by TEOH SHEN KHANG and submitted as partial fulfillment of the requirements for the degree of Doctor of Philosophy (Engineering) at Universiti Tunku Abdul Rahman.

Approved by:

Humaira

(Prof. Ts. Dr. Humaira Nisar) Date: 30 July 2023
Supervisor
Department of Electronic Engineering
Faculty of Engineering and Green Technology
Universiti Tunku Abdul Rahman

(Dr. Yap Vooi Voon) Date: 27 July 2023
External Co-supervisor
Department of Computer Science
Aberystwyth University

FACULTY OF ENGINEERING AND GREEN TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 27 July 2023

SUBMISSION OF THESIS

It is hereby certified that **TEOH SHEN KHANG** (ID No: **16AGD06100**) has completed this thesis entitled "**A SCENE INVARIANT CONVOLUTIONAL NEURAL NETWORK FOR VISUAL CROWD COUNTING USING FAST-LANE AND SAMPLE SELECTIVE METHODS**" under the supervision of Prof. Humaira Nisar (Supervisor) from the Department of Electronic Engineering, Faculty of Engineering and Green Technology, and Dr. Yap Vooi Voon (External Co-Supervisor) from the Department of Computer Science, Aberystwyth University.

I understand that the University will upload softcopy of my thesis in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(TEOH SHEN KHANG)

DECLARATION

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

Name	Teoh Shen Khang
Date	27 July 2023

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Adagrad	Adaptive Gradient
ADAM	Adaptive Movement Estimation
CCNN	Counting Convolutional Neural Network
CNN	Convolutional Neural Network
CUDA	Compute Unified Device Architecture
DESA	Department of Economic and Social Affairs
ELU	Exponential Linear Unit
FL	Fast Lane
GT	Ground Truth
HOG	Histogram Oriented Gradients
IE	Inference Engine
LReLU	Leaky Rectified Linear Unit
LSTM	Long-Short Time Memory
MAE	Mean Absolute Error
MO	Model Optimiser
MRE	Mean Relative Error
MSE	Mean Square Error
NLP	Natural Language Processing
OpenVINO	Open Visual Inference and Neural network Optimization
ReLU	Rectified Linear Unit
RMSProp	Root Mean Square Propagation
SiCNN	Scene Invariant Convolutional Neural Network
SS	Sample Selective
SVM	Support Vector Machine
Tanh	Hyperbolic Tangent
VGG	Visual Geometry Group

CHAPTER 1

INTRODUCTION

1.1 Background

Visual surveillance systems are becoming common and their deployment is becoming more widespread as societies become more complex and the population continues to grow. The Population Division of Department of Economic and Social Affairs (DESA) of the United Nations predicts that the world population will reach 8.5 billion in 2030 and 9.7 billion in 2050. Crowd can often be seen at airport, bus station, tourist attraction and public display. These are illustrated in Figure 1.1. The study of crowd analysis has received great attention from researchers recently. The issue of public security practice has arisen with the exponential development of the world population, thereby resulting in more frequent crowd gatherings in the recent years. In such scenarios, it is essential to analyse crowd behaviour for better management, safety and security. The essential part of crowd control is crowd counting and it has piqued the interest of many researchers.

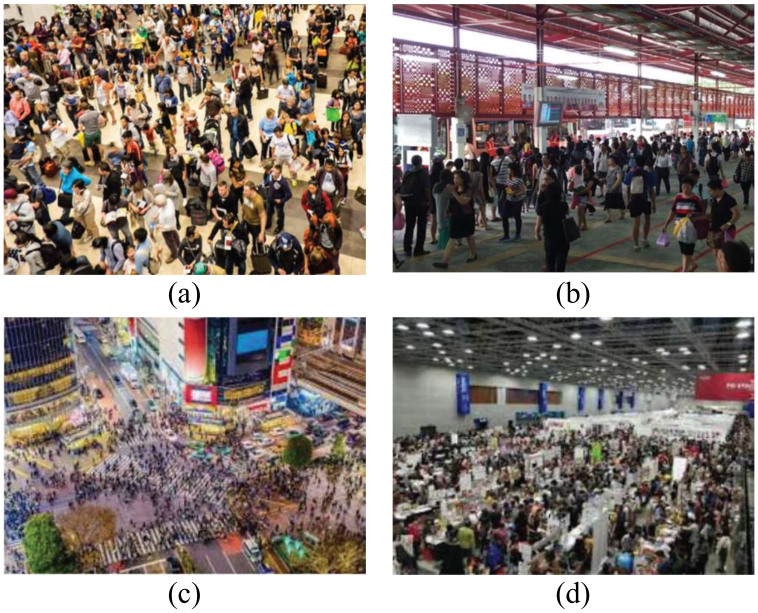
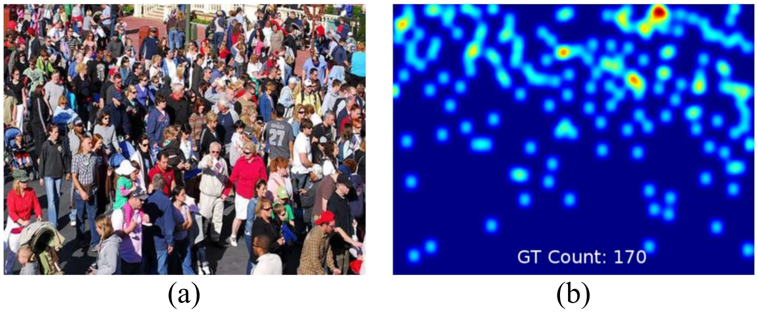


Figure 1.1 : Illustration of various crowd scenes. (a) Airport (b) Bus terminal (c) Tourist spot (d) Public exhibition. Occlusions, static and dynamic human objects and different perspective can be observed from the image.

Crowd counting and density estimation aims to count and estimate the number of pedestrian or the density of crowds in a monitored area. Like any other computer vision problem, crowd counting presents many challenges, such as occlusions, static and dynamic object, uneven distribution of people and uneven lighting, which complicates matters considerably (Sindagi and Patel, 2018; Khan, Menouar and Hamila, 2023). Some of the challenges are illustrated in Figure 1.1. With the implementation of Artificial Intelligence into the visual surveillance, automated crowd counting received much attention in management of crowds for social safety.

Early research on crowd counting focused primarily on detection-based approaches. This method aims to locate the exact position of each human object in the scene. Head detection and face detection is widely used trained detector to locate human object. However, these approaches did not work well in a complex scene such as perspective changes, different lighting and strong occlusion. Subsequently, researchers proposed regression-based approaches to either regress the number of human objects straight from the image or to create a density map to predict the crowd size. A crowd density map illustrates the spatial arrangement of the individual from the image that generated with Gaussian kernel. It is often used in density estimation, where an image of a crowd is mapped onto the corresponding density map, which indicates the number of people per pixel on the image (as illustrated in Figure 1.2).



(a) (b)

Figure 1.2 : Illustration of density map estimation (Sindagi and Patel, 2018). (a) Input image (b) Corresponding density map with count.

With the significant improvement of the convolutional neural network (CNN) in solving computer vision tasks, many researchers are motivated to exploit the potential to estimate the crowd size (Khan et al., 2020). CNN-based crowd counting methods can be generally categorised into density map prediction and global regression. In practice, the method for generating the density maps is crucial for crowd counting. Improperly generated density maps may dramatically affect the counting performance. The typical design of density map prediction algorithm is divided into two steps. First, the ground-truth density maps of crowd images are generated from the ground-truth dot maps (density map generation) by convolving with a Gaussian kernel. Second, deep learning models are designed to train on the generated density map and predict a density map from an input image. The crowd size is then estimate from the predicted density map. However, there are few issues in this methodology.

According to (Wan and Chan, 2019), the Gaussian kernel bandwidth parameters used in the density map generation method are selected manually. The parameters of the kernel bandwidth or kernel shape used to generate the density map are often dataset dependent and such setting usually do not work across different dataset. Extra effort is needed to manually select the parameters again when working at another dataset. Another issue in the density map methodology is the quality of estimated crowd density maps. Many existing CNN-based approaches have several max-pooling layers in their networks compelling them to regress on down-sampled density maps. Lastly, most methods optimise over traditional Euclidean loss which is known to have certain disadvantages. Regressing on down-sampled density maps using Euclidean loss results in low quality density maps and therefore decrease the accuracy (Fan et al., 2022).

To address this issue, crowd estimation using global regression could provide better performance in term of estimation accuracy and execution speed compared to density map. Global regression can skip a step by estimating the final count directly from the images, whereas density estimation requires first predicting a density map, which is then summed to obtain the final count. Moreover, ground truth annotation to generate the density map is highly labour-intensive. This step cannot be skipped before the deep learning model training process can take place. According to (Ma et al., 2021), the entire annotation process of UCF-QNRF dataset involved 2,000 human-hours to its completion merely for 1,535 images.

Although significant progress has been made, existing methods typically follow the training-testing protocol within a single dataset and suffer from significant cross-data performance degradation (Ma et al., 2021). These works are scene-specific, that is, the models learned for a particular scene can only be applied to the same scene. Accuracy drops significantly when the models are applied to unseen datasets. Given an unseen scene, the model must be re-trained with new annotations. The poor generalisability of existing crowd counting models has seriously restricted their applications in the real scenario. Therefore, a scene invariant model that can be applied to various scenes and reduce deployment cost is highly motivated.

1.2 Application of CNN-based Crowd Counting Techniques

This section presents the possible applications of convolutional neural network-based crowd counting methods. These techniques covered a diverse range of applications include intelligent crowd analysis, health-care applications, disaster management and public-event management.

Crowd-counting techniques can be beneficial when used to gather information for intelligent analysis and conclusions. For instance, the queue length in front of a restaurant reception or billing reception. The management could observe and analyse this information accordingly to optimise the number of staff members, thus reducing the labour cost. Waiting times at traffic lights could be optimised in terms of traffic flow, especially during rush hours. In addition, appropriate product placement in large shopping centres and shops can be done by analysing the crowd heat map.

Crowd counting techniques hold a vital role in health-care systems (Sohail et al., 2021). It became significantly important when come to cancer early-stage diagnosis where it is important to count a number of cancerous cells especially with patients suffering from cancer (Litjens et al., 2016). Moreover, (Andre et al., 2019) presented a deep learning architecture for histopathologic cancer diagnosis to increase the objectivity and efficiency of histopathology-slide analysis.

Crowd counting techniques can be applied in public exhibition, concert and sports event to count the number of people. Therefore, these events can be managed safely by counting and analysing the crowd to prevent disastrous situations. These techniques are also had advantages in managing available resources such as optimising crowd movements and spatial capacity to minimise the life-threatening event when a portion of the crowd panics and charges in random directions resulting in huge numbers of people have died from suffocation in highly crowded areas. Therefore, early detection of overcrowding and better crowd management can be made possible by utilising and deploying the crowd counting techniques in the surveillance system.

1.3 Motivation and Research Objectives

As described in section 1.1, the CNN-based density map estimation model reaches its limits when trained with an improperly created density map, and the quality of the predicted density map is caused by multiple max-pooling layers. Model optimisation over Euclidean loss also resulting in low quality density map prediction. Furthermore, existing methods usually applied the training-testing protocol within a single dataset or scene specific. The counting performance decreases drastically when the models are applied to unseen scene. It is important to address this issue for better generalisability and reduce the restriction to deploy the model in the real scenario. This study focused on originate a crowd counting CNN model using global regression methodology with effective training strategy to count the crowd in different unseen scene.

The objective of this study is to design a convolutional neural network for crowd counting based on whole image-based inference methodology. This inference methodology benefited in reducing the training time. Second objective is to extend the capability of the proposed convolutional neural network when estimate the crowd size in unseen scene. The last objective is to optimise the performance of the proposed convolutional neural network through novel model training strategy. Experiments will be conducted in three datasets where one dataset is used for training and another two datasets are treated as unseen scene to the model.

1.4 Thesis Overview

There are five chapters in this thesis. Chapter 1 gives a short introduction to the overview, applications, strengths and weaknesses of the crowd counting model. The motivation and research objective are included in the last section of chapter 1. Chapter 2 contains the literature survey of crowd counting methods and techniques. Chapter 3 focuses on the methodology used in the design of Scene Invariant CNN (SiCNN) model. This chapter explains the algorithm, technique, training strategy and dataset developed for this study. Chapter 4 provides the results and discussions on the experiments conducted to evaluate the SiCNN model performs in single and multi-different scene. The inference time of the proposed model is also evaluated in edge embedded platform for real time performance. Chapter 5 summarises and concludes the finding of this study. Future recommendations for future works are also given in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter starts with a review of existing crowd counting algorithm that involved the traditional approaches and CNN approaches. Its pros and cons are discussed next. After that, the inference methodology for training the model is reviewed. Lastly, the public available crowd datasets are also analysed and discussed.

2.2 Traditional Crowd Counting Approaches

As crowd counting received more attention, the challenge of estimating the crowd size from the image using computer vision has been addressed and approached from different angles. In general, the crowd counting method can be mainly classified into the following categories: detection-based approaches, regression-based approaches and CNN-based approaches. Comprehensive review on CNN-based approaches is conducted in this section. Since this study focused on CNN-based approaches, for the sake of completeness this section briefly discusses detection-based and regression-based approaches with hand-crafted features.

2.2.1 Detection-based Approaches

Early research for crowd counting focussed on detection framework. A human detector is used to acquire the approximate location of each people in the scene and the detected humans are count together to indicate the crowd size. This framework usually performed detection in either whole body or partial body. Whole body detection approaches generally are hand-crafted detection methods which train a classifier such as Support Vector Machine (SVM), boosting and random forest to detect human features such as Haar wavelet features(Viola and Jones, 2004), histogram oriented gradients (Dalal and Triggs, 2005), edgelet (Wu and Nevatia, 2005) and shapelet (Sabzmeydani and Mori, 2007). Figure 2.1 illustrates the whole body detection approaches to determine the crowd size. Although these methods have proven effective to some degree in low crowd scenes, they are compromised by the presence of high density crowds.

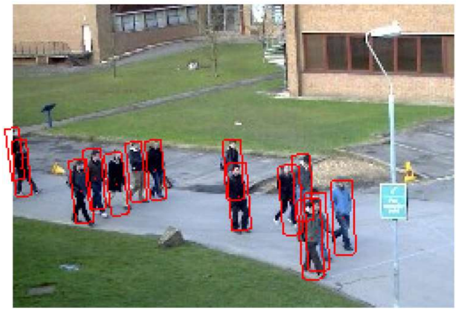


Figure 2.1 : Illustration of fully body detection. There are total of 14 pedestrians in the image. (Ryan, 2010)

Researchers have attempted to address the limitation of full body detection issue by adopting part-based (partial body) detection approaches (Wu and

Nevatia, 2007; Felzenszwalb et al., 2010). The classifiers are boosted to detect specific body parts, for instance the head and shoulder. The detected part is then sum together. This method is slightly better than whole body detection. Among the literatures, head detection and face detection are widely used. The authors (Wu and Nevatia, 2007; Keju, Fuqiang and Zhipeng, 2009) trained a face detector to identify the faces of the pedestrians when they entered the camera's view and estimate the crowd size. Figure 2.2 illustrates the face detection method for crowd density estimation. These approaches have limitation when the face is not visible to the camera. As a summary, the detection-based approaches are mainly engaged by early researchers. These methods have drawbacks in crowded and occluded environment where the human parts are not fully visible for detection (Saleh, Suandi and Ibrahim, 2015).

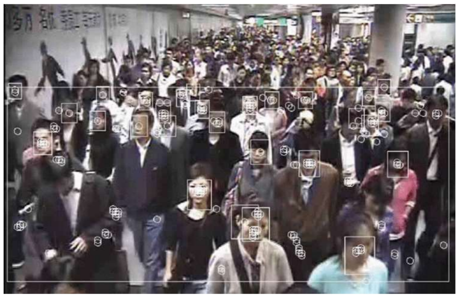


Figure 2.2 : Illustration of face detection. The classifier identified a total of 25 pedestrian from the image. (Keju, Fuqiang and Zhipeng, 2009)

2.2.2 Regression-based Approaches

The part-based detection approaches were used to reduce the issues of occlusion; however, these methods were not effective in the presence of extremely dense crowds (Sindagi and Patel, 2018). To address these issues, researchers proposed regression-based approaches where they attempted to regress the count by learn a mapping between features extracted from image to the total number of people. The benefit of counting with regression is to avoid the relatively complex human detection process and avoid heavily rely on the learning classifier.

The regression methodology contains two major components. The first component is low-level feature extraction. The features extraction involved global features and local features. Global features include foreground features extracted from the video using typical background subtraction technique and blob-based features such as perimeter and area perimeter ratio. Local features include texture such as histogram oriented gradients (HOG), size such as motion segment and edge/gradient features. These features have been used to further improve the results. The second component is regression model. Once the low-level features are extracted, a variety of regression models such as neural network (Kong, Gray and Tao, 2006), linear regression (Ryan, 2010) and support vector regression (Percannella et al., 2010) are used to learn the mapping from these global and local features to the number of people. (Ryan, 2010) mapped the total number of foreground pixels and the number of people in the scene approximately linear. Gaussian process regression is used to obtain the crowd density by model the relationship between foreground pixels and crowd size. Figure 2.3 illustrate the regression approach in estimating the crowd size.

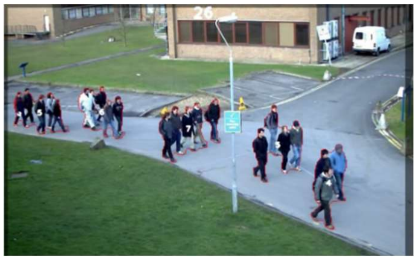


Figure 2.3 : Illustration of regression method using blob as local feature and Gaussian process regression as learning model to estimate the crowd size. (Ryan, 2010)

2.3 CNN-based Crowd Counting Methods

With the advent of deep learning and the success of CNNs in countless computer vision tasks, researchers have used the nonlinear features of CNNs for learning to estimate the corresponding density maps or counts. A diverse of CNN-based techniques have been proposed by the researchers. According to the training dataset used by the network and the output of the network, the CNN based crowd counting techniques can be generally categorised into detection-based CNN and regression-based CNN. This section will broadly review these methods and further categorise them based on their network architecture and training pattern. Their strengths and limitations are review in this section.

A detection-based CNN technique aims to accurately locate and count the objects in the image. (Stewart, Andriluka and Ng, 2016) introduced an end-to-end training network structure. Using GoogLeNet, each image was transformed into 1024-dimensional features, containing valuable location information. Long short-term memory (LSTM) (Van Houdt, Mosquera and Nápoles, 2020) was employed to map these features to a series of detection boxes, ordered by decreasing confidence. When LSTM fails to find any detection box with confidence exceeding the predetermined threshold, the process stops. Crowd size is determined by summing all the bounding boxes. This method is suitable for situations where detection objects have minimal overlap, but its effectiveness diminishes significantly in high-overlap scenarios. (Li et al., 2020) proposed "HeadNet", an adaptive relational network designed to extract context information and mitigate missed human head detection. They employed Resnet-101 as the feature extraction network to process input images and utilized local structured feature modules to enhance individual stability. The network then generated bounding boxes for each detected head, ultimately determining the crowd size.

Among the CNN-based model, regression-based CNN methods are widely used for count and density estimation (Sindagi and Patel, 2018; Patwal et al., 2023). This method is trained utilising an image dataset through either unsupervised methods or being annotated by point (density map). During testing, the network directly output the total number of people in the image or generates the density map corresponding to the image. According to the training dataset used by the network and the output of the network, these methods are divided into two categories: regression density map method and regression counting method. The regression density map methods train the network to predict a density map and then estimate the crowd density from it. In contrast, the regression counting method outputs the number of people in an image directly. Regardless of which regression method, the architecture of CNN crowd counting can be subcategorised into basic CNN model and scale-aware CNN model.

Basic CNN model can be regarded as the pioneer of deep-learning methods for density estimation that can be applied to obtain a crowd count in real time due to the uncomplicated network architecture. CNN-based crowd counting was first exploit by (Wang et al., 2015). The authors adopted the AlexNet network (Krizhevsky, Sutskever and Hinton, 2017) to design an end-to-end CNN regression model for total count estimation from extremely dense crowd. They replaced the last fully connected layer of 4096 neurons of AlexNet network with a single neuron layer to predict the crowd size. In order to increase the prediction accuracy and reduce non-related background such as buildings and tress in the images, the authors added additional negative samples where the ground truth number was set to zero to enrich the training data.

Instead of estimating the total count, (Fu et al., 2015) adopted Multi-stage ConvNet (Sermanet, Chintala and Lecun, 2012) to classify the image into five classes of crowd density. These classes are very high density, high density, medium density, low density and very low density. The multi-stage network has better shift, scale and distortion invariance. Additional classifier is used to boost the accuracy by reclassify the rejected samples. In (Hu et al., 2016), the authors proposed a deep-learning approach to estimate mid-level and high-level crowds in an image. A regressor was used to estimate the number of individuals in a local area, while the total density was estimated by summing the estimated densities of the local regions. In their work, a convolutional neural network architecture is learned to estimate crowds. (Zhao et al., 2016) proposed a CNN model to count the number of people crossing a line in surveillance videos. (Walach and Wolf, 2016) enhanced their CNN model with layer boosting and removing negative samples to decrease processing time and improve counting accuracy. Layer boosting works by increasing the number trained network layers to iteratively train a classifier that is used to fix the errors of the previous one. Negative sample method is to minimise the impact of low-quality data during model training process.

In contrast to the above methods that predict a density map, (Shang, Ai and Bai, 2016) proposed an end-to-end count estimation method using CNN architecture. The method takes the entire image as input and directly outputs the final number of people. Thus, computations in overlapping regions are performed jointly by combining multiple processing stages, reducing complexity. The network simultaneously learns to estimate local counts, and contextual information is integrated into the network to ignore background noise for better performance. Figure 2.4 shows the architecture is designed with a pre-trained GoogleNet model, a Long-short time memory (LSTM) decoders, fully connected layers and a single regressor node for the final count. The image contains crowd is input to the network the high-dimensional CNN feature maps are computes using the GoogleNet network. The feature maps are then decoded into local count using the LSTM unit. The local counts are map to the fully connected layers and the final count is output to the single regressor node.

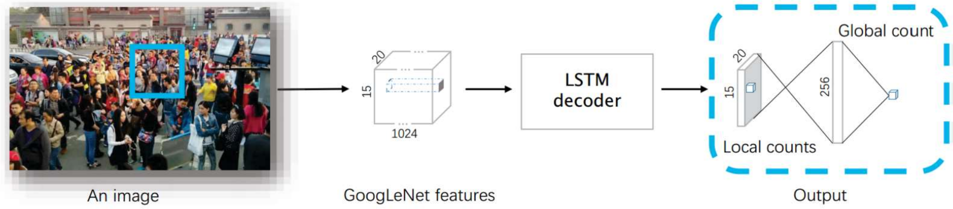


Figure 2.4 : Overview of the end-to-end counting method using CNNs. (Shang, Ai and Bai, 2016)

Scale-aware CNN models are evolved from basic CNN models into more sophisticated models that were robust to variations in scale. (Zhang et al., 2016) proposed a multi-column network with different convolution kernels size to predict a density map and the crowd estimation is performed next. Multi-column network is widely used for image recognition before it is started to apply in crowd counting. The proposed method ensures robustness to large variations in object size by constructing a network of three columns corresponding to filters with different sized receptive fields. The multi-column network is shown in Figure 2.5.

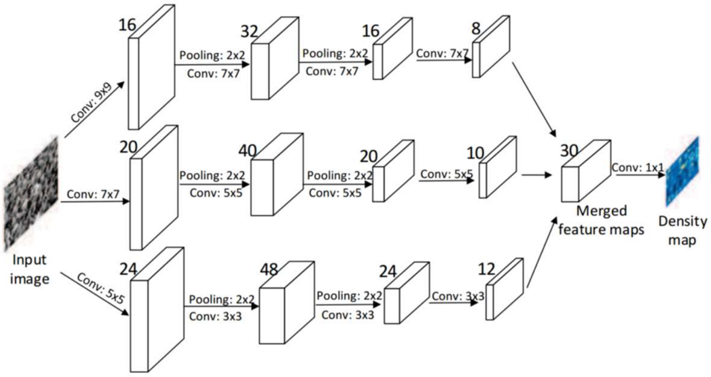


Figure 2.5 : Overview of single image crowd counting via multi-column network. (Zhang et al., 2016)

Inspired by the above approach, (Oñoro-Rubio and López-Sastre, 2016) first employed a deep fully convolutional neural network with six convolutional layers called Counting CNN (CCNN) as illustrated in Figure 2.6. The network is then extended into Hydra CNN that consists of 3 heads and a body with each head learning features for particular scale. Each head is constructed using the CCNN model, whose outputs are combined and passed to the body, which consists of two fully-connected layers. The Hydra CNN is able to estimate object densities in a variety of crowded scenarios without explicit geometric information of the scene. While the different heads extract image descriptors at different scales, the body learns a high-dimensional representation that merges the information provided by the heads at multiple scales. The final product is a 18x18 density prediction. The crowd size is determined from the density map.

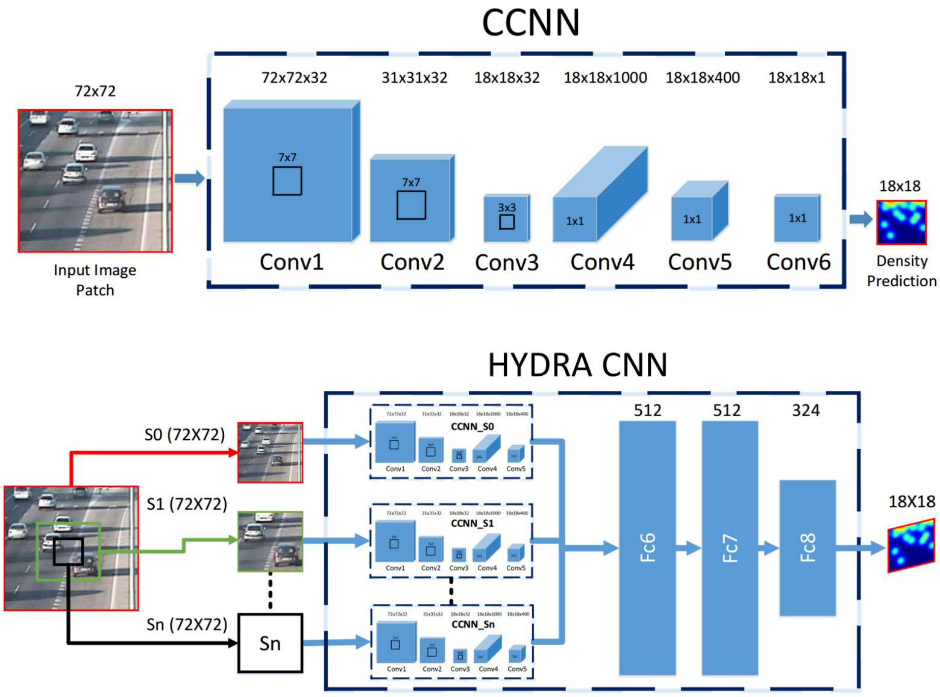


Figure 2.6 : Overview of Hydra-CNN network. (Oñoro-Rubio and López-Sastre, 2016)

(Marsden et al., 2017a) made observation on the earlier scale aware models are difficult to optimise and are computationally complex. The authors proposed a single column fully convolutional network (Figure 2.7) to incorporate scale information using an effective and straightforward multi-scale averaging step during prediction. The number of people is estimated for each scale and the final number is the average of all estimates. The training set in this work is contrast to the early methods that uses highly overlapping cropped patches. This technique constructed four image quadrants to ensure no overlap and avoid potential overfit when the network is continuously trained on the same set of pixels. Therefore, the generalisation performance of the network is improved.

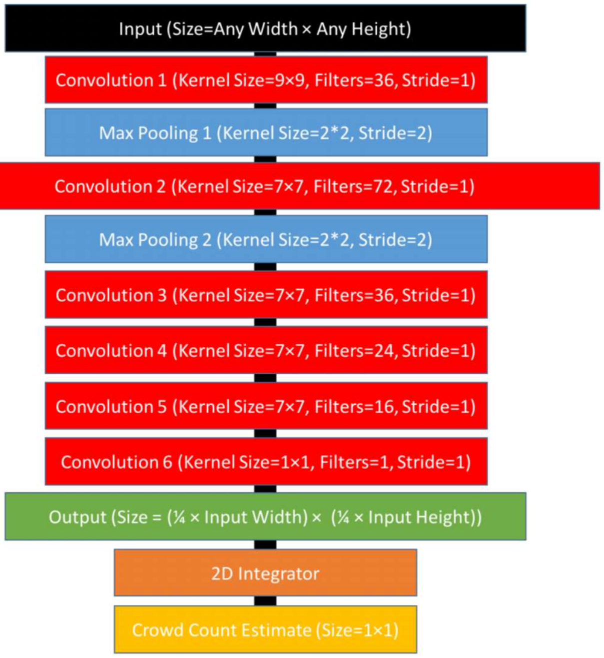

Input (Size=Any Width × Any Height)
Convolution 1 (Kernel Size=9×9, Filters=36, Stride=1)
Max Pooling 1 (Kernel Size=2*2, Stride=2)
Convolution 2 (Kernel Size=7×7, Filters=72, Stride=1)
Max Pooling 2 (Kernel Size=2*2, Stride=2)
Convolution 3 (Kernel Size=7×7, Filters=36, Stride=1)
Convolution 4 (Kernel Size=7×7, Filters=24, Stride=1)
Convolution 5 (Kernel Size=7×7, Filters=16, Stride=1)
Convolution 6 (Kernel Size=1×1, Filters=1, Stride=1)
Output (Size = (¼ × Input Width) × (¼ × Input Height))
2D Integrator
Crowd Count Estimate (Size=1×1)

Figure 2.7 : Overview of Fully Convolutional Network for crowd counting. (Marsden et al., 2017a)

2.4 Cross Scene Crowd Counting Methods

State-of-the-art crowd counting deep learning model usually fine-tune the pre-train deep neural network like AlexNet to determine the crowd size. Despite a lot of effort has been made, current approaches usually follow the training-testing protocol within a single dataset and suffer from significant cross-dataset performance degeneration. In fact, the accuracy drops drastically when models are applied to unseen datasets or unseen scene. Models that did not have good generalisability are hardly to deploy their applications in the real scenario.

The crowd counting methods described in the section 2.2 and section 2.3 are scene-specific that the model learned for a particular scene can only work well in the same scene. The methods usually do not take a strategy to narrow the gaps between images and do not generalise well to unseen scene. The researchers (Zhang et al., 2015; Ma et al., 2021; Zhang et al., 2023) analysed existing methods and found that their performance drops sharply when applied to a new scene that is different from the training dataset. Given the difficulty of training deep networks for new scenes, it would be important to explore how to benefit from models trained on existing sources. Most existing methods retrain their models for a new scene, which is not practical in real-world scenarios as it would be expensive to obtain annotations for each new scene. To the best understanding when this research work commences, there are limited research works to address the issue.

(Zhang et al., 2015) He tried to solve this problem by performing data-driven training without the need for labelled data for new scenes. Their technique learns a mapping of images to crowds and adapts this mapping to new target scenes for cross-scene counting. To achieve this, they optimised the network to alternately train on crowd counting and density estimation to achieve better local optima. Their network is adapted to new target scenes without the need for additional labelling information. In an another approach, (Liu and Vasconcelos, 2015) attempted to infuse transfer learning into crowd counting. A model adaptation technique for Gaussian process counting was introduced. The technique implementing the source model as a prior and the target dataset as a set of observations, the components are combined into a predictive distribution that captures information in both the source and target dataset. However, the idea of data driven and transfer learning for crowd scenes are relatively unexplored and is a nascent area of research.

2.5 CNN Training Methodology

Another factor that influences the prediction accuracy of the CNN model is the training methodology. The CNN training process is an essential part of building a robust network. The training loss is calculated during the training process and it will be used to update the network weights. Training methodology can be generally classified into patch-based training and whole image-based training. These methodologies can be used to improve the prediction accuracy of the network or the quality of its density map.

2.5.1 Patch-based Training Methodology

CNN model trained in patch-based methodology utilise patches cropped from the image and a sliding window that runs over the test image (Datta Gupta et al., 2023). Figure 2.8 illustrates the patch-based training architecture. This method is useful in applications where the resolution quality of the density map that cannot be ignored and it needs to enhance. For instance, in cancer diagnosis that both the affected cell count and the resolution of affected cells are essential information. As a result, this method is widely used to enhance and predict an accurate density map for crowd size estimation. The drawback for patch-based training is high computational cost due to every image need to process by the small sliding window (Sindagi and Patel, 2018). The reviewed literatures discussed in section 2.3 are using patch-based methodology to train their respective network to predict a crowd density map and eventually determine the crowd size.

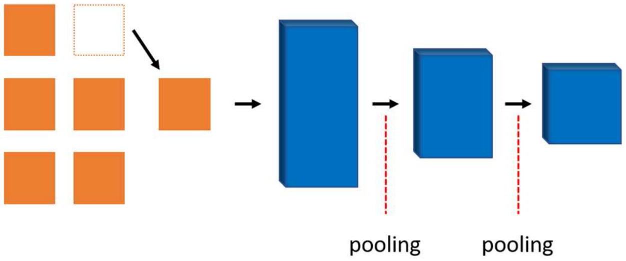


Figure 2.8 : Architecture of patch-based training methodology. (Ilyas, Shahzad and Kim, 2020)

2.5.2 Whole Image-based Training Methodology

In contrast to patch-based training methodology, whole image-based training methodology inputs an entire image into the CNN model and directly predicts the final count (Wang et al., 2023). Figure 2.9 illustrates the whole image-based training architecture. This training pattern minimises the network computational cost which are very useful in real-time applications. These techniques have applications in pedestrian counting, tracking person of interest and analyse passing pedestrian across CCTV. (Zhang, Shi and Chen, 2018) proposed a scale-adaptive CNN architecture with a backbone of fixed small receptive field. The network was trained on relative count loss and density map loss using whole image-based training approach to improve the network generalisation on crowd scenes. The authours (Marsden et al., 2017b) He proposed a multi-criteria method using residual deep learning to study crowd counting, violence detection and density classification. This kind of architecture can be generally named as multi-task CNN model. (Wilie, Cahyawijaya and Adiprawita, 2018) redefined the crowd counting process by using a Xception network and fully connected layers. The Xception network is a pre-trained parameter used as transfer learning to learn to count. The fully connected layers are connected to a single node to predict the crowd size. Other than applying whole image-based training methodology in crowd counting, a CNN-based fruit-counting technique was proposed by (Rahnemoonfar and Sheppard, 2017) by using a deep-simulated-learning algorithm. The network was trained on synthetic data and a modified version of the Inception-ResNet architecture (Szegedy et al., 2017) was used to count the fruits (tomatoes). The models discussed in this section typically have limitation in missing of negative sampling and lack of data-driven approach.

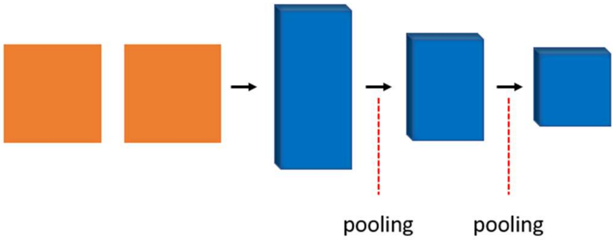


Figure 2.9 : Architecture of whole image-based training methodology. (Ilyas, Shahzad and Kim, 2020)

2.6 Activation Function in CNN

In convolutional neural network literature, there is a significant interest in identifying and defining activation functions which can improve neural network performance. Activation function helps the neural network to use important information while suppressing irrelevant data points during the feedforward propagation. In feedforward propagation, the activation function is a mathematical "gate" between the input to the neuron and its output to the next layer. When looking at any neural network architecture, activation function is one of the essential elements that generally applied in the hidden layer or the output layer.

The activation function can be generally classified into linear activation function and non-linear activation function. This section focuses on reviewing the non-linear activation functions instead of the linear activation function, as the linear activation function does not allow the model to create complex mappings between the inputs and outputs of the network (Nwankpa et al., 2018). A crowd image often contained complex scene and various illumination that the linear activation function is not able to perform the mapping.

This section discussed six non-linear activation functions that most commonly used in recent years and their respective illustrations are shown in Figure 2.10. They are sigmoid activation function, hyperbolic tangent (Tanh) activation function, rectified linear unit (ReLU) activation function, leaky ReLU (LReLU) activation function, Exponential Linear Unit (ELU) activation function and Swish activation function.

The sigmoid activation function has a S-shape graph that takes any real value as input and outputs values in the range of 0 to 1 (Ding, Qian and Zhou, 2018). The larger the input, the closer the output value to 1.0, whereas the smaller the input, the closer the output value to 0.0. The sigmoid activation function is commonly used for models to predict probability as an output due to the probability of anything exists only between the range of 0 and 1. The sigmoid activation function is defined as follows:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

where x is the input to the activation function. The limitation is that the sigmoid activation function binds a large range of inputs to a small range between 0 and 1. Therefore, it always produces a non-negative value as output and a large change to the input value leads to a small change to the output value, resulting into small gradient values as well.

The Tanh activation function is very similar to the sigmoid activation function, which has the same S-shape, but differs in the output range from -1 to 1 (Ding, Qian and Zhou, 2018). In Tanh, the larger the input, the closer the output value to 1.0, whereas the smaller the input, the closer the output value to -1.0. The tanh activation function inherits all the valuable properties of the sigmoid activation function where the formula is defined as follows:

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \tag{2.2}$$

where x is the input to the activation function. Based on the function, the range of possible outputs are expanded and zero-centered which includes negative, positive and zero outputs. Hence, the output values can be easily mapped as strongly negative, neural, or strongly positive. Tanh activation function is computationally expensive because of the exponential function. Similar to sigmoid activation function, it binds a large range of input to a small range between -1 and 1. Thus, a large change to the input value leads to small change to the output value.

Since the proposal by (Fukushima, 1975), the rectified linear unit (ReLU) activation function has been widely used in convolutional neural network because of its efficient properties. The ReLU activation function is defined as:

$$f(x) = \max(0, x) \tag{2.3}$$

where x is the input to the activation function. Since ReLU is not an exponential function and it forces negative values to zero, it has low computational cost. These features make ReLU a better choice to be used in CNN as it provides better performance and generalisation when compared to the sigmoid and tanh function. However, the consequence of forcing the negative inputs to zero is the network will suffer from dead neuron problem. Therefore, the improved version of ReLU, called LReLU and ELU were introduced to solve the dead neuron problem.

The LReLU activation function is very similar to the ReLU function, with the difference that it has a small positive slope in the negative range (Ding, Qian and Zhou, 2018). The LReLU is defined as follows:

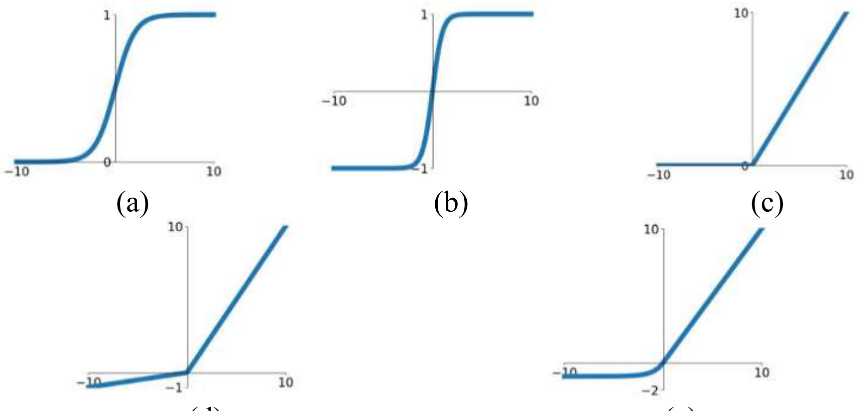
$$f(x) = \begin{cases} 0.01x & for\ x\ \leq 0 \\ x & otherwise \end{cases} \tag{2.4}$$

where x is the input to the activation function. Unlike ReLU, the LReLU activation function did not force the negative inputs to zero and allow to be passed on as outputs. Therefore, the dead neurons are no longer encountered. The drawback of this function is the prediction may not be consistent for negative input values because of the 0.01 derivative.

ELU activation function is a variant of ReLU that modifies the slope of the negative part of the function that uses a log curve to define the negative values (Ding, Qian and Zhou, 2018). The ELU is defined as follows:

$$f(x) = \begin{cases} x & for\ x\ \leq 0 \\ \alpha(e^{x}-1) & for\ x\ > 0 \end{cases} \tag{2.5}$$

where x is the input to the activation function and α is the constant that defines function smoothness when inputs are negative. ELU is a strong alternative to ReLU because of the features to avoid dead neuron problem by introducing log curve for the negative values of input and helps the network nudge weights and biases in the right direction. The limitation of the ELU activation function is high computational time because of the exponential function. The illustration of the discussed activation function as graphs are shown in Figure 2.10.



(a) (b) (c)

(d) (e)

Figure 2.10 : The graph depiction of non-linear activation functions ((Nwankpa et al., 2018); Ding, Qian and Zhou, 2018)). (a) Sigmoid (b) Tanh (c) ReLU (d) LReLU (e) ELU

2.7 Weight Initialisation in CNN

This section describes and discusses the weight initialisation strategies for convolutional neural network. Weight is the parameter that part of the neuron set that consists of inputs and a bias value. As an input enters the node, it gets multiplied by a weight value and the resulting output is often passed to the activation function (described in section 2.6). The activation function will determine the activation of the neuron, that is, to pass the multiplied value to the neuron output. Weight initialisation is a crucial step to implement before training any neural network. The weights of a network are initialised that define the starting point and then adjusted/optimised repeatedly while training the network. The weight initialisation directly drives the convergence of a network where the loss converges to a minimum value and an ideal weight matrix is obtained. Therefore, the selection of an appropriate weight initialisation strategy becomes important for end-to-end training. Three weight initialisation strategies are selected from the literature and will be reviewed in this section. They are random weight initialisation, Xavier weight initialisation and He weight initialisation.

Random weight initialisation is a strategy to randomly assigns random values except for zeros as weights to neuron paths (Narkhede, Bartakke and Sutaone, 2022). It was introduced in an attempt to overcome the limitation of zero weight initialisation. The zero weight initialisation assigned zero as initial value to the weights and it is highly ineffective as neurons learn the same feature during each iteration in the model training. The random weight initialisation strategy tries to address the problems of zero weight initialisation since it prevents neurons from learning the same features of their inputs. This strategy ensures the neuron learn different functions of its input and gives much better accuracy than zero weight initialisation. However, assigning values randomly is highly prone to overfitting and vanishing gradient problem.

The current standard strategy to initialise the weights of neural network layers is called Xavier weight initialisation (Glorot and Bengio, 2010). In Xavier weight initialisation, the weights are initialised in such the way that the variance of the activations are the same across every layers. This constant variance is calculated by takes in the number of fan in (number of input paths towards the neuron) and the number of fan out (number of output paths towards the neuron) into account to help prevent the gradient from exploding or vanishing. The Xavier weight initialisation strategy is designed to work well with sigmoid or tanh activation function.

Xavier weight initialization performs effectively with sigmoid or tanh activation functions but demonstrates suboptimal performance when used with ReLU activation functions. The Xavier weight initialisation will cause the activations start to collapse to zero at the deeper layer of the neural network and caused no learning. To address this issue, a modified version of the strategy was developed specifically for ReLU activation. It is named as Kaiming weight initialisation or He weight initialisation (Kaiming et al., 2018). The Kaiming weight initialisation strategy is computed as a random number with a Gaussian probability distribution with a mean of 0.0 and a standard deviation of sqrt(2/n), where n is the number of input paths to the neuron. This strategy will make the weights healthier for the ReLU as the mean of weight should have slightly incremented layer by layer. More neurons will get activated and the neural network learn better. Therefore, The Kaiming weight initialisation strategy is designed to work well with ReLU activation function.

2.8 Gradient Descent Algorithm

Gradient descent is a standard optimisation algorithm frequently applied to train deep learning network and to minimise the cost function. A gradient is a mathematically measurement to quantify the direction of the ascent or descent of a line or curve and descent is the action of going downwards. In short, the gradient descent is the algorithm that facilitates the search of parameters values (weights and biases) that minimise the cost function towards a local minimum or optimal accuracy (Mustapha, Mohamed and Ali, 2020). It captures the local slope of the function, allowing the network to predict the effect of taking a small step from a point in any direction. It is one of the parameters to update the network's weight. Three gradient descent algorithms are reviewed in this section. They are Adaptive Gradient algorithm (Adagrad), Root Mean Square Propagation (RMSProp) algorithm and Adaptive Movement Estimation (ADAM) algorithm.

Adagrad algorithm (Duchi, Hazan and Singer, 2011) is an algorithm for gradient-based optimisation. It is a stochastic optimisation method that adapts the learning rate to the parameters. It performs smaller updates for parameters associated with frequently occurring features and larger updates for parameters associated with infrequently occurring features. For this reason, it is suitable for dealing with sparse data, for example native language processing (NLP) and image recognition. Each parameter has its own learning rate that improves performance on problems with sparse gradients. One of the benefits of using Adagrad algorithm is that it eliminates the need to manually tune the learning rate. Moreover, the network optimised using Adagrad algorithm is converge faster and more reliable. The limitation of the Adagrad algorithm is that the learning rate will starts to shrink end eventually become significantly small due to the accumulation of the squared gradients in the denominator(Sun et al., 2020). Extremely small learning rate will cause the network no longer able to learn additional knowledge.

RMSProp algorithm is an unpublished optimisation algorithm designed for neural network proposed by Geoff Hinton (Vitaly Bushaev, 2018). The RMSProp algorithm tries to resolve Adagrad's radically diminishing learning rates by using a moving average of the squared gradient. In RMSProp, the learning rate gets adjusted automatically and it chooses a different learning rate for each parameter. It divides the learning rate by the average of the exponential decay of squared gradients.

ADAM algorithm is different to classical stochastic gradient descent where the stochastic gradient descent maintains a single learning rate for all weight updates and the learning rate does not change during training (Kingma and Ba, 2015). Instead, ADAM is an adaptive learning rate optimisation algorithm that utilises both momentum and scaling. The author combines the advantage properties of Adagrad and RMSProp to tackle the sparse gradients and noise gradients. Instead of adapting the parameter learning rates based on the average mean as in RMSProp, the ADAM algorithm makes u se of the average of the uncentered variance of the gradients. Specifically, the algorithm calculates an exponential moving average of the gradient and the squared gradient, and it control the decay rates of these moving averages. Due to ADAM algorithm is computationally efficient, has very little memory requirement and able to achieve good results fast, it is one of the popular algorithms in the field of deep learning (Sun et al., 2020).

2.9 Datasets

In recent decades, a large number of datasets have been created, allowing researchers to develop models with better generalisation capabilities. The datasets contain images of low-density crowds, while the most recent ones focus on high-density crowds, which poses numerous challenges for the researcher, such as scale variations, clutter and heavy occlusion. The creation of these datasets has motivated approaches to evolve from traditional methods to deep learning methods. In this section, three publicly available key datasets are reviewed.

2.9.1 **Mall Dataset** (Chen et al., 2012)

The Mall dataset was collected by Chen using a surveillance camera installed in a mall. The dataset has different lighting conditions and people densities. In addition to the different density levels, there are also different activity patterns such as static and dynamic crowds. In addition, the dataset is challenging because it is heavily obscured by the objects in the scene, such as a houseplant or a stall located along the walkway. The dataset consists of 2000 individual images of size 320x240 with 6000 marked pedestrians. When creating the dataset, the first 800 images were used for training and the remaining 1200 images for the evaluation. This dataset has the density crowd with the lowest of 13 people in the image and the highest of 53 people in the image.

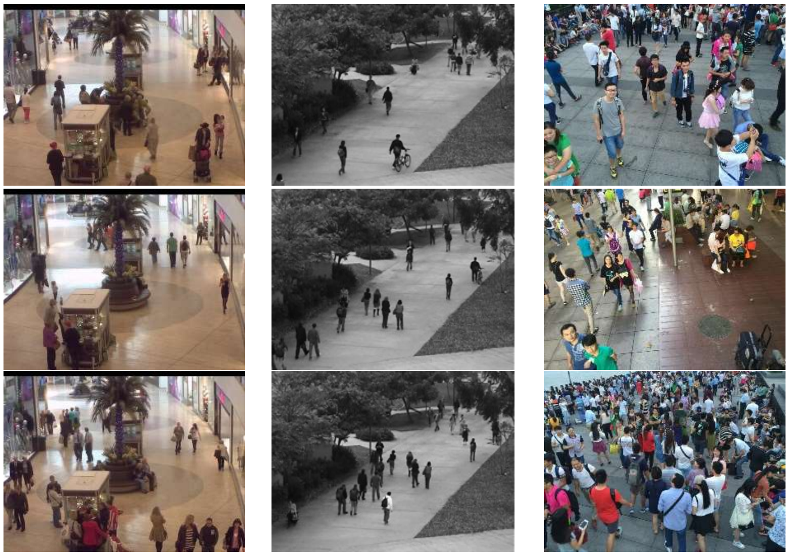
2.9.2 **UCSD Dataset** (Chan, Liang and Vasconcelos, 2008)

The UCSD dataset was originally created for anomaly detection but later it is used by the researcher for counting people. The dataset was captured by a camera on an outdoor pedestrian walkway. The dataset consists of 2000 frames of a video sequence together with ground truth annotations. A region of interest is provided to ignore unnecessary moving objects. The dataset contained a total of 49,885 pedestrian annotations. The setting of the dataset used the first 800 frames for training and the remaining 1200 frames used for testing. This dataset has the density crowd with the lowest of 11 people in the image and the highest of 46 people in the image.

2.9.3 **Shanghai Tech Dataset** (Zhang et al., 2016)

Zhang et al. presented a new large-scale crowd counting dataset consisting of 1198 images with 330,165 labelled pedestrians. The dataset is one of the largest in terms of the number of labelled pedestrians. It consists of two parts: part A and part B. Part A contains a total of 482 images randomly selected from the internet. Part B consists of images taken on the streets of major cities in Shanghai. Part A has a much higher density compared to part B. The setting of part A has 300 frames for training and the remaining 182 frames used for testing. The setting of part B has 400 frames for training and the remaining 3316 frames used for testing. The crowd in part A is extremely congested scenes randomly chosen from the Internet while part B includes relatively sparse-crowd scenes taken in different scenes from the streets of metropolitan areas in Shanghai city. The dataset successfully attempts to create a challenging dataset with diverse scene types and varying density levels. part A has the density crowd with the lowest of 33 people in the image and the highest of 3139 people in the image, respectively, whereas part B has the density crowd with the lowest of 9 people in the image and the highest of 578 people in the image, respectively.

Sample images from the three datasets are shown in Figure 2.11. It can be observed that the UCSD and the Mall dataset have relatively low density images and less variations in illuminations. The Shanghai Tech dataset has significant variations in the density levels and different perspectives across images.



(a) (b) (c)

Figure 2.11 : Sample images from various datasets. (a) Mall (b) UCSD (c) Shanghai Tech.

2.10 Discussion

A variety of CNN-based methods have been investigated in section 2.2, section 2.3 and section 2.4. These methods can be generally categorise based on property of the networks and training approach. The categories for the network architecture are basic CNNs and scale-aware CNN models. Furthermore, the CNN training methodology can be classified into patch-based methodology and whole image-based methodology.

The techniques fall under the basic CNNs category are mainly focus on density estimation instead of crowd count. However, recent research on using a single regressor node at the last year enable basic CNNs to predict the actual count instead of estimating the crowd density. Approaches such as negative sample and data-driven are missing in this category. The speed of training and inferencing can be enhanced by removing redundant samples. By iteratively reducing errors in different network, error-rate probability can also be reduced.

The techniques fall under scale-aware CNNs category are mainly focus on predicting a density map and the crowd size is determine from the density map next. The key issue of this category is the quality of estimated crowd density maps. Many approaches have a number of max-pooling layers in their network compelling them to regress on down-sampled density maps. Next, most methods optimise over traditional Euclidean loss which is known to have certain disadvantages. Regressing on down-sampled density maps using Euclidean loss results in low quality density maps and therefore decrease the accuracy. Moreover, ground truth annotation to generate the density map is highly labour-intensive. This step cannot be skipped before the deep learning model training process can take place. As described in section 1.1, the entire annotation process of a big dataset is a high labour cost and it has the possibility to prone to error.

Patch-based methodology is useful in enhancing the resolution quality of the density map by using a sliding window to process through the image. However, the sliding window processing is the main cause in high computational cost due to many small patches need to process in each image. This methodology is mainly used in post-processing rather than in real-time processing. In contrast, whole image-based methodology inputs an entire image into the CNN model and directly predicts the final count. This technique is not widely used in density map prediction due to less resolution quality. It is observed by the researchers that by using the whole image for inferencing and training, it results in reduction of complexity as the computations shared on overlapping regions. Therefore, the whole image-based methodology minimises the network computational cost which are very useful in real-time applications.

As described in section 2.4, most of the current approaches in crowd estimation did not work well when the model is tested with unseen dataset. Specifically, the models usually follow the training-testing protocol within a single dataset and the model's accuracy suffer from the unseen dataset that is not trained by the model. The models have to retrain on every new scene and it is impractical due to expensive training cost. To date, there are limited research works to address the issue. Therefore, this study is motivated to design a deep learning model that can perform the crowd counting in scene invariant where the model only needs to train once.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, the details on the design of CNN-based architecture for scene invariant crowd counting are discussed. The designed network chosen for this research is named as Scene Invariant CNN (SiCNN). The model's name is motivated by the literatures discussed in section 2.10. The methodology in designing the model will be discussed in detail in their respective sections later in this chapter. The SiCNN consists of a single backbone network with multiple residual blocks attached to it. The details for the design of the backbone network are described in section 3.2. Section 3.3 presents the methodologies used to train the proposed model. Section 3.4 covers the evaluation metrics used to evaluate the proposed model. The dataset used for training and testing the model is discussed in section 3.5. Section 3.6 and section 3.7 presented the strategies used to improve the counting accuracy. The results of each section will be discussed in chapter 4.

3.2 Single Backbone Network – Scene Invariant CNN (SiCNN)

Many state-of-the-art works adopted the multi-column architecture with different filter sizes to generate competitive density maps. The crowd size estimation is determined from the density map. As mentioned in section 2.10, multiple max pooling layers in the network and model training with Euclidean distance decreases the quality of the density map leads to accuracy dropped. Therefore, this research instead designed a Scene Invariant CNN (SiCNN) where the model uses a single backbone network with a single filter size. The full architecture of SiCNN is illustrated in Figure 3.1. The implementation details are explained in the subsequent parts.

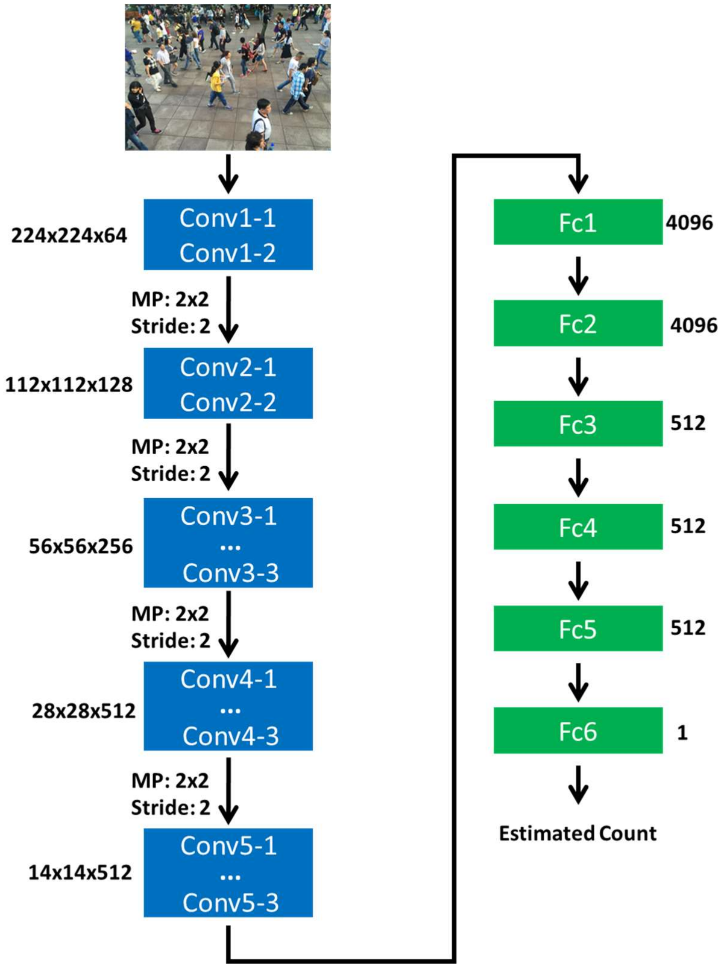


Figure 3.1 : The structure of the Scene invariant Convolutional Neural Network (SiCNN) for crowd counting. The convolutional layer's parameters are represented as "Conv(layer number) (filter size x filter size x filter number)". The fully connected layer's parameters are denoted as "Fc(layer number) (number of neurons)". MP: max pooling layer.

The SiCNN model is designed based on the inspiration of the Visual Geometry Group architecture or better known as VGG-16 (Simonyan and Zisserman, 2015), which is a convolutional neural network model proposed by A. Zisserman and K. Simonyan from the University of Oxford. The VGG-16 is a standard deep CNN architecture with 16 convolutional layers and a groundbreaking model that achieved top-5 accuracy in ImageNet competition where the dataset consisting of more than 14 million images belonging to 1000 classes. Instead of using large kernel size filters, the model replaced it with several 3x3 kernel size filters, thereby reducing the computational time and complexity of the model. Part of the SiCNN model is motivated by the VGG-16 architecture due to its powerful generalisation capability. Generalisation of the CNN is ability to perform unseen data. It is an essential point to a crowd counting model that can handle different scene.

The architecture of SiCNN model is forged with 13 convolutional layers and 6 fully connected layers. The backbone design followed the power of two network topology. Matrix multiplication is one of the central computations in deep learning. CPU and GPU are operating in Single Instruction Multiple Data (SIMD) (Choi and Lee, 2021) to process the data and information. Since the design of the physical processors are often a power of two, the SiCNN model is designed with power of two network topology can align properly with the number of physical processors. For example, a GPU with 512 physical cores can fit a fully connected layer with 512 neurons and each unit is able to process parallelly in one cycle instead of process individually. Therefore, the model training time can be reduced. The number of filter channel in the convolutional layers are configured as 64, 128, 256 and 512 sequentially. The units in the fully connected layers are configured as 4096 and 512 sequentially.

The main processing in the convolutional neural network is convolution that basically a dot product of kernel or filter and patch of an image (local receptive field) of the same size. During the model training or learning process of CNN, different kernel sizes or filter sizes will affect the accuracy and the training time. In early research work, larger filter size such as 11x11 and 5x5 are widely used in image recognition and classification. In general, large filter size would reduce more noise from the image. However, at the same time, the larger filter will result in a loss of image detail (Camgozlu and Kutlu, 2020). Moreover, large filter size also increases the trainable parameters and cause the model training time to increase. For example, AlexNet (Krizhevsky, Sutskever and Hinton, 2017) CNN architecture was introduced in 2012, it used several 11x11 and 5x5 filter sizes that consumed more than two weeks in training resulting in extremely large number of hyper-parameters to be trained and expensiveness. The SiCNN model instead focused on one single 3x3 filter size and one single maxpool layer size of 2x2 with stride 2.

By studying the analysis of filter size in deep learning (Khanday, Dadvandipour and Lone, 2021) the 3x3 filter size produces better results in term of accuracy. This research work only utilised small filter size. First, most of the useful features in an image are usually local and it makes sense to take few local pixels at a time to apply convolutions. Second, these features may be found in more than one place in an image. Sliding a single small filter size all over the image have advantage in extracting the useful features in different parts of the image. Lastly, small filter size added benefit of weight sharing and reduction in computational costs. Small filters preserve the spatial resolution of the input to enable building deeper network. “Deeper” has similar effect as “Wider” (multi-column architecture) does in a network.

To enhance the regularisation of the model and avoid overfitting, batch normalisation or commonly abbreviated as Batch Norm technique (Ioffe and Szegedy, 2015) is applied to the neurons’ output as pre-activation before applying the activation function. This technique normalises activations in a network across the mini-batch of definite size. For each feature, batch normalisation computes the mean and variance of that feature in the mini-batch. It then subtracts the mean and divides the feature by its mini-batch standard deviation. A neuron with Batch Norm can be generally computed as follows:

$$z^N = \left(\frac{z - m_z}{s_z}\right) \tag{3.1}$$

where z^N is the output of Batch Norm, z is the output of neuron before Batch Norm, m_z is the mean of the neuron’s output and s_z is the standard deviation of the neuros’ output.

With the properties of Batch Norm, the research work exposed the intuitions of the most important reasons. Firstly, this technique normalised the layer’s inputs by re-centering and re-scaling into a similar range of values, thus speed up the learning. Secondly, in the original paper (Ioffe and Szegedy, 2015) observed that Batch Norm reduces the internal covariate shift of the network. The internal covariate shift is a change in the input distribution of an internal layer of a Neural Network. Applying Batch Norm ensures that the mean and standard deviation of the layer inputs will always remain the same, thus the amount of change in the distribution of the input of layers is reduced. It eventually benefits the model can be trained stably.

Activation function is one of the core elements in neural network. A neural network without an activation function will essentially act as a linear regression model. The activation function adds the non-linear transformation to the input making it capable to learn and perform more complex tasks. It generally applied in the hidden layer or the output layer. The primary role of the activation function is to transform the summed weighted input from the node into an output value to be fed to the next hidden layer or as output. After analysed the literatures of the non-linear activation function (described in section 2.6), ReLU and ELU activation functions are chosen and applied to the SiCNN model. ReLU (Rectified Linear Units) activation is added to the convolutional layers whereas ELU (Exponential Linear Units) is added to the fully connected layers. ReLU has a derivative function and allows for backpropagation while simultaneously making it computationally efficient (Dubey, Singh and Chaudhuri, 2022). Since the neurons with positive values will only be activated, useful features from the image can be effectively highlighted.

Despite the fact that the ReLU activation are widely used in both the convolutional layers and the fully connected layers, this research work observed dying ReLU issue (described in section 2.6) in which the deactivated neurons make the gradient value zero and caused the weights are not updated. Subsequently decreases the model's ability to train from the data properly. Instead, this problem is solved by utilised ELU (Clevert, Unterthiner and Hochreiter, 2016) activation in the fully connected layers. ELU is a strong alternative for ReLU to avoid dead ReLU problem by introducing log curve for negative values of input. The ELU function helps the SiCNN model to have more flexibility when estimating the final count. The last fully connected layer is consisting of one neuron denoted as regression node to predict the final crowd size.

3.3 SiCNN Model Training Methodology

The proposed designed model is trained using the whole image-based training pattern with a whole image is treated as input to the model, as illustrated in Figure 3.1. The whole image-based training pattern is described in section 2.5.2. The designed SiCNN model is trained in an end-to-end manner from scratch. By default, the network's weights are initialised with small random numbers. These weights are used in the neuron to calculate a weighted sum of the inputs. Having a good weight initialisation prior to the model training can increase the model optimisation. The weight initialisation strategies are discussed in section 2.7. This research chosen Kaiming weight initialisation strategy to initialise the network weights because it works well with ReLU and ELU activation function. The Kaiming method is calculated as a random number with a Gaussian probability distribution. It gives a good range of constant variance for ReLU and ELU activation.

As discussed in section 2.8, ADAM optimisation algorithm is computationally efficient and yet able to achieve good results fast when compared to Adagrad and RMSProp. Thus, the model is trained (gradient descent) using the Adaptive Movement Estimation (ADAM) optimisation algorithm. It is a replacement optimisation algorithm for stochastic gradient descent for training deep learning models. The algorithm can operates using less memory but still maintain efficiency. Intuitively, ADAM combines the best properties of the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp) algorithm that can handle sparse gradients on noisy problems. Furthermore, learning rate decay policy can also be used with ADAM. Generally, it is often useful to reduce learning rate as the training progresses during the model training. This method will prevent the model to skip the global minimum. The model training applied the exponential decay policy where the learning rate starts from 1e-4 and decays to 1e-6.

To prevent the model from overfitting, a standard model validation technique named as k-Fold cross-validation is applied. This technique split the dataset into k equal parts. The first part will label as testing test and the remaining part will label as training set. The model will be trained based on this setting. In the second iteration, the 2nd part will label as testing test and the remaining part (including the first part) will label as training set. The iteration will go on until all the parts are involved as training/testing set. Each iteration will return an accuracy score and the final score is the average of the summed score. Five-fold cross-validation is chosen in this research work. Finally, the model is trained with a batch size of 5 and 200 epochs in total. Mean Absolute Error (MAE) loss function is adopted to measure the loss between the ground truth and the estimated count. The formula is defined as follows:

$$MAE = \frac{1}{N}\sum_{1}^{N}|y_i - y_i'| \qquad (3.2)$$

where N is the number of test data, y_i is the ground truth and y_i' is the predicted result corresponding to the i^{th} data. Further information on MAE is discussed in the following section.

3.4 Evaluation Metrics

Typically, the deep learning model trained to estimate the crowd density or directly predict the crowd size is classified as a regression model. The standard evaluation metrics used to evaluate a regression model are the Mean Absolute Error (MAE) and the Mean Square Error (MSE). These metrics are convenient mechanisms for evaluating the amount of deviation between the ground truth and the predicted values and they are widely used by the researchers, in particular in crowd counting (Saleh, Suandi and Ibrahim, 2015; Sindagi and Patel, 2018; Fan et al., 2022). A factor contributing to the widespread utilisation of these metrics is their simplicity in computation and typically having a low level of computational complexity. Furthermore, these metrics are usually used as benchmarks for comparison between CNN-based crowd-counting models and are also used to rank the model. The MAE and MSE are defined as:

$$MAE = \frac{1}{N}\sum_{1}^{N}|y_i - y_i'| \tag{3.3}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{1}^{N}(y_i - y_i')^2} \tag{3.4}$$

where N is the number of test data, y_i is the ground truth and y_i' is the predicted value corresponding to the i^{th} data.

MAE represents the average of the absolute difference between the ground truth and the predicted values in the dataset. MSE represents the average of the squared difference between the ground truth and predicted values in the dataset. In general, MAE denotes the accuracy of a visual crowd counting and MSE denotes the robustness of a visual crowd counting. Lower score of MAE and MSE indicate that the model has good counting performance. The proposed designed SiCNN model is evaluated with MAE and MSE so that the results can benchmark with other research works.

Despite the fact that the MAE and MSE are the standard evaluation metrics for crowd counting model, this research discovered that the metrics did not give concrete performance of a model. The metrics did not measure the behaviour of the model in predicting the size of a crowd. For example, a crowd counting model achieved a score of 2 MAE and a score of 10 MSE, the scores only reveal the accuracy of the model's predictions and do not show whether the model consistently overestimates or underestimates the crowd size or if there is a balance between these two scenarios. It is a good indicator to include into one of the network properties. Therefore, this study employed an additional evaluation method called error rate distribution to assess the model's performance in crowd counting. This method is described in section 4.3.

3.5 Datasets

In the world of deep learning, a dataset is a collection of data used to train the model. A dataset is used as an example to teach the deep learning algorithm how to make predictions. The proposed designed SiCNN model is validated on three different public available crowd counting datasets: ShanghaiTech dataset, UCSD dataset and Mall dataset.

The ShanghaiTech dataset (Zhang et al., 2016) consists of 1198 annotated images with a total of 330,165 person. The number of annotated persons per image is range from 33 to 3,139. This dataset is split into two parts: ShanghaiTech part A and ShanghaiTech part B. The crowd in ShanghaiTech part A is extremely congested scenes randomly chosen from the Internet while ShanghaiTech part B includes relatively sparse-crowd scenes taken from the streets of metropolitan areas in Shanghai city. This research work selected ShanghaiTech part B for validation instead due to the multiple different scenes and it fulfil the objective of this research work. Following the training methodology explained in section 3.3, the ShanghaiTech part B applied five-fold cross-validation technique for model training and model testing. The dataset successfully attempts to create a challenging dataset with diverse scene types and varying density levels. Some samples are shown in Figure 3.2. The validation results are discussed in chapter 4.

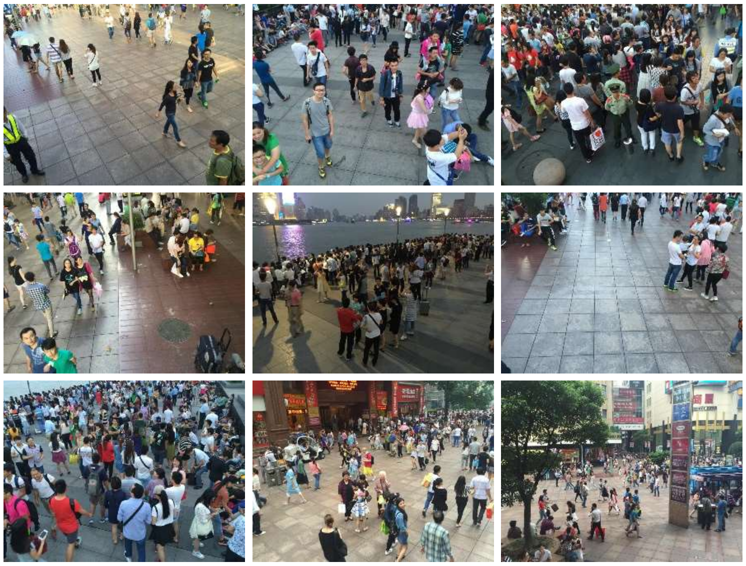


Figure 3.2 : Samples for the ShanghaiTech part B dataset. They consist of different scenes with medium to high density crowd.

The UCSD dataset (Chan, Liang and Vasconcelos, 2008) was among the first datasets created to count people. It consists of 2,000 frames taken from a fixed video camera at a pedestrian walkway. The dataset contains a total of 49,885 pedestrian instances and is divided into a training set and a test set. While the training set contains 800 images, the test set contains the remaining 1200 images. The number of persons is from 11 to 46 per frame. As discussed in section 2.9.2, the dataset was captured from a single location and there are no differences in the perspective of the scene between the images. Some samples are shown in Figure 3.3.

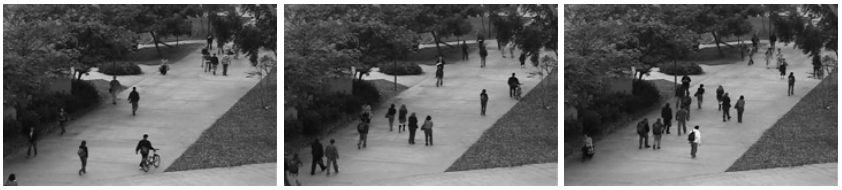


Figure 3.3 : Samples for the UCSD dataset. It was captured on a single static camera.

Mall dataset was collected by (Chen et al., 2012) with diverse illumination conditions and crowd densities from a surveillance camera installed in a shopping mall. Along with having various density levels, it also has different activity patterns for instance static and moving crowds. Additionally, the scene contained in the dataset has severe perspective distortion resulting in large variations in size and appearance of objects. The dataset has a total of 2000 frames of size 320 x 240 with 6000 instances of labelled human objects. In comparison to the UCSD dataset, the Mall dataset has relatively higher crowd density images. However, as discussed in section 2.9.1, the datasets do not have any variation in the scene perspective across images. Some samples are shown in Figure 3.4.

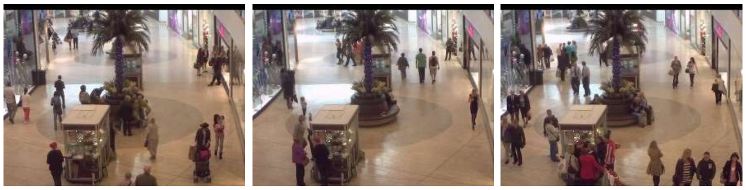


Figure 3.4 : Samples for the Mall dataset. The images come from the same video sequence providing no variation in perspective across images.

The proposed designed SiCNN model is trained and validated on ShanghaiTech part B where it is rich of different crowd scenes. UCSD dataset and Mall dataset contain only single scene which will be used to validate the cross-scene crowd counting performance of the SiCNN model. It is worth mentioning that the SiCNN model did not train with UCSD dataset and Mall dataset. These two datasets are act as an out of sample or unseen data to SiCNN model. The cross-scene performance is discussed in chapter 4.

3.6 Fast Lane Method

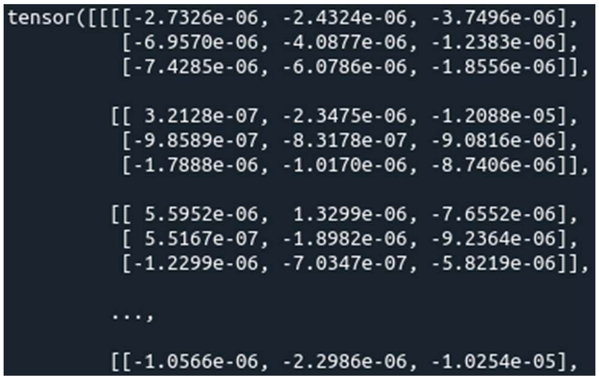
Although deep CNN can represent very complex functions and excel in solving computer vision task, it was addressed by (He et al., 2016) there is a huge barrier to obtain the optimal local minimum due to degradation of the network. Therefore, this research work conducted investigation for the SiCNN training process and discovered that the model has vanishing gradients and zero weight update. These issues are causing gradient descent prohibitively slow. The gradient descent formula can be generally defined as follows:

$$W_{new} = W_{old} - lr * \frac{d}{dW} f(x) \tag{3.5}$$

where W is the neuron's weight. lr is the learning rate and the $\frac{d}{dW} f(x)$ is the gradient.

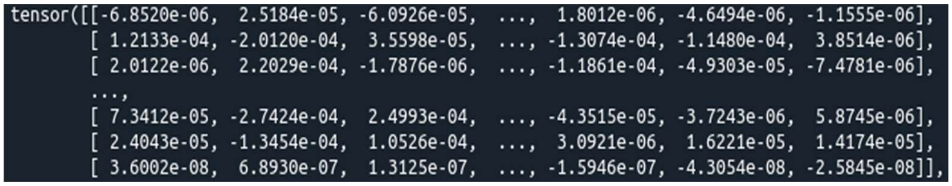
More specifically during gradient descent, the back propagation from final layer back to the first layer will multiply the weight matrix with the gradient. If the gradients are too small for multiple multiplication, the gradient can decrease exponentially quickly to zero. For instance, using equation 3.5 as a reference, assume the gradient has a value of 1x10^{-6}, the W_{old} has a value of 10.

The W_{new} value would be literally unchanged due to the gradient is significantly small. This problem results in vanishing gradient and the weight is not change. As the model iterates eventually, it will not converge to a global optimum. Figure 3.5 and Figure 3.6 shown the gradient from the one of the convolutional layers and fully connected layer respectively. It can be observed that the gradients are generally range between 10^{-5} and 10^{-7}. They are relatively small and potentially caused the model hardly to train well.



```
tensor([[[[-2.7326e-06, -2.4324e-06, -3.7496e-06],
          [-6.9570e-06, -4.0877e-06, -1.2383e-06],
          [-7.4285e-06, -6.0786e-06, -1.8556e-06]],

         [[ 3.2128e-07, -2.3475e-06, -1.2088e-05],
          [-9.8589e-07, -8.3178e-07, -9.0816e-06],
          [-1.7888e-06, -1.0170e-06, -8.7406e-06]],

         [[ 5.5952e-06,  1.3299e-06, -7.6552e-06],
          [ 5.5167e-07, -1.8982e-06, -9.2364e-06],
          [-1.2299e-06, -7.0347e-07, -5.8219e-06]],

         ...,

         [[-1.0566e-06, -2.2986e-06, -1.0254e-05],
```

Figure 3.5 : Gradient values from one of the convolutional layers.



```
tensor([[-6.8520e-06,  2.5184e-05, -6.0926e-05,  ...,  1.8012e-06, -4.6494e-06, -1.1555e-06],
        [ 1.2133e-04, -2.0120e-04,  3.5598e-05,  ..., -1.3074e-04, -1.1480e-04,  3.8514e-06],
        [ 2.0122e-06,  2.2029e-04, -1.7876e-06,  ..., -1.1861e-04, -4.9303e-05, -7.4781e-06],
        ...,
        [ 7.3412e-05, -2.7424e-04,  2.4993e-04,  ..., -4.3515e-05, -3.7243e-06,  5.8745e-06],
        [ 2.4043e-05, -1.3454e-04,  1.0526e-04,  ...,  3.0921e-06,  1.6221e-05,  1.4174e-05],
        [ 3.6002e-08,  6.8930e-07,  1.3125e-07,  ..., -1.5946e-07, -4.3054e-08, -2.5845e-08]],
```

Figure 3.6 : Gradient values from one of the fully connected layers.

CNN can be a powerhouse for major machine learning algorithm. Typically, stack more layers to the network, that is, going deeper or increasing the depth could increase the performance because more number of neurons are available to learn the abstract features. Instead of increasing the number of layers which cause the network degradation, this research work improved the

SiCNN model to a wider network by introduced fast lane connections to link between the layers. This technique is inspired by the gated shortcut connections of Highway Network (Srivastava, Greff and Schmidhuber, 2015). The idea is origin from Long Term Short Memory recurrent network. These networks allow unimpeded information flow across many layers on information highways.

In traditional neural network, each layer feeds into the next layer. The goal of the fast lane connection is to feed the output of the layer directly into the layer about few hops away. The algorithm allows the network to learn an identity function. The layers with fast lane connection are named as residual block. Figure 3.7 illustrated the fast lane connections in general.

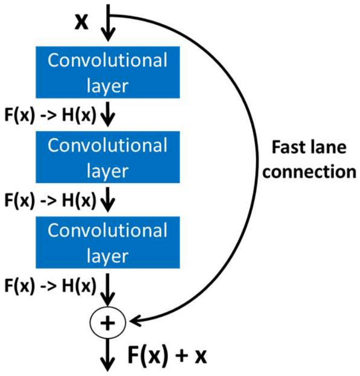


Figure 3.7 : The diagram shown the network with fast lane connection (residual block). The connection is skipping three layers and the x value is summed with the output from the 3rd convolutional layer before input to the next layer.

In Figure 3.7, the network is trying to learn the correct mapping, i.e. F(x) -> H(x), where x is the input, H(x) is the expected output and F(x) is the network try to fit to resembles H(x). With the fast lane connection, the equation of the residual block is updated as follows:

$$H(x) = F(x) + x \tag{3.6}$$

It has been observed that it is easier to learn residual of output and input, rather than only the input. Adding a large number of fast lane connection could increase the execution time that leads to slow prediction. Therefore, to strike the balance between accuracy and execution speed, the SiCNN model is enhanced by adding 1 fast lane connection to the convolutional layers and 2 fast lane connections to the fully connected layers. The new model is shown in Figure 3.8.

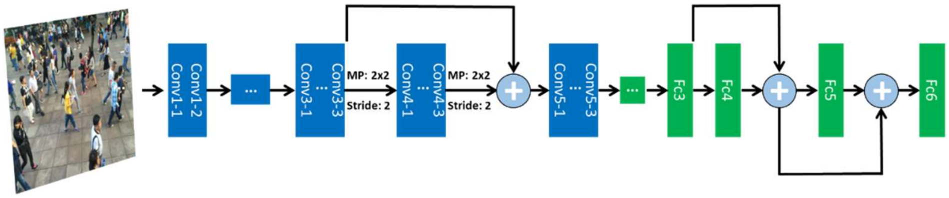


Figure 3.8 : The SiCNN model with fast lane connections.

The first fast lane connection is connected from the output of conv3-3 layer and combined with the output of conv4-3 before connects to conv5-1. This fast lane connection is named as a convolutional block where it consists of a convolutional layer of 512 channel of 3x3 kernel, batch normalisation and ReLU activation function. The convolutional block is illustrated in Figure 3.9.

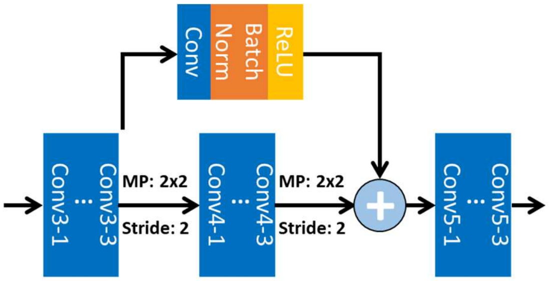


Figure 3.9 : Convolutional block.

There are two additional fast lane connections in the fully connected layers. The first lane is connected from the output of Fc3 to the input of Fc5. The second lane is connected from the combined output to the input of Fc6. Unlike the convolutional block, which connects different convolutional layer dimensions, the fast lane connections in the fully connected layers link every input neuron to every output neuron, creating a single dimension without any dimension issues. With the new designed fast lane connection added to the model, the learned features from the earlier layers can be propagated to the later layer and avoid the information become too abstract to learn. Larger gradients from the initial layers can also propagate to the deep layer and beneficial for the model convergence. It is worth noting that training time is reduced because wider models take advantage of GPUs being more efficient in parallel computations. The new result is discussed in chapter 4.

3.7 Sample Selective Method

A dataset serves as an example to teach the algorithm how to make predictions. Selecting the correct dataset is one of the crucial steps of successfully developing a quality deep learning model. Other than mitigate the network degradation, this section performed investigation on the dataset participation in the model training.

Based on observations, there were samples where the model accurately or closely estimated the crowd size during the early stages of its training process. Such samples are labelled non-essential samples. Allowing these samples to remain involved in the training even after the model has learned to make accurate predictions can introduce bias into the network, which can negatively impact its generalisation performance. Moreover, continuing to train on would also consume unnecessary training resources. Furthermore, the investigation revealed that the presence of outlier samples is another factor that decreases training efficiency. Samples are referred to as outliers when the difference between the predicted crowd size and the actual crowd size is excessively large. These samples will not benefit the model's performance and it will increase the training time.

Inspired from (Wang et al., 2015) work where the authors have enriched the training data with negative samples, whose ground truth count is set to zero. Essentially, the authors removed negative samples from the model training to improve the accuracy of crowd estimation. In this study, a sample selective method is formulated to reduce the impact of non-essential samples and outlier samples by periodically reducing their participation in the training process. The algorithm engages the absolute difference to identify the sample's quality. An absolute difference is the distance between two numeric values; the ground truth and the predicted crowd size, disregarding whether is a positive value or negative value. A low distance value between the ground truth and the predicted value indicates a non-essential sample, while a high distance value indicates an outlier.

To determine the non-essential sample and outlier sample, two thresholds denote as prediction error, t_{tri} and t_{out} are selected. Based on the initial cross validation tests performed on 30% of the training data of the ShanghaiTect part B dataset, a t_{tri} of 35 and a t_{out} of 150 are properly determined. The training data that did not meet the threshold criteria are temporary removed from the training process for several epochs. Specifically, these samples "sleeps" for three times in the entire 200 epochs. The "sleeps" duration is five epochs. The sample selective algorithm is activated at epoch number 50, 100 and 150. Figure 3.10 illustrates the sample selective activation time. During the sample selective activation time, the model is train on the samples without the non-essential samples and outlier samples. Conversely, when it is not within the sample selective activation period, the model is trained using the entire training dataset. From the observation, an average of 25% active samples are removed during the sample selective stage.

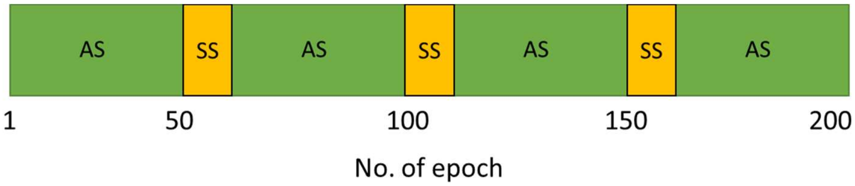


Figure 3.10 : Sample selective arrangement on the training data. AS: All Samples. SS: Sample Selective.

In contrast to (Wang et al., 2015) work where they permanently removed the negative samples to improve the accuracy, resulting in not all sample from the dataset are participated in the model training. This may create bias that their model only good in predicting the crowd size using good sample and it may not train with sufficient data. In contrast, the sample-selective algorithm performed a temporal elimination of samples rather than permanently removing them from the model training. This temporal elimination of noisy training samples has clear advantages in terms of training time and accuracy. The new result is discussed in chapter 4.

3.8 Summary

In this chapter, a Scene Invariant Convolutional Neural Network (SiCNN) model for visual crowd counting is introduced. The model is designed with single backbone network methodology. It consisted of 13 convolutional layers and 6 fully connected layers. Optimisation techniques such as power of two network topology, single size filter, batch norm, ADAM optimiser and k-fold cross-validation are applied to increase the training accuracy. Three datasets are applied in the model training. ShanghaiTech part B dataset is used to train the model while UCSD dataset and Mall dataset are treated as unseen scene to evaluate the model's cross-dataset performance.

Vanishing gradient and zero weight update issues are found during the model training. These issues are addressed and solved by the fast lane connection method where this technique brought large gradients from the initial layers to the deep layer to improve the model convergence. After conducted careful experimental work, three fast lane connections are attached to the SiCNN model. The participation performance of the dataset is evaluated during the model training stage, revealing low-quality samples. These samples include non-critical samples, where the model is able to accurately or closely predict the correct output during the early stages of its training process, and outlier samples, where the model's prediction deviates significantly from the actual output. Such samples waste training resources and are not beneficial to the model. Therefore, a sample selective algorithm is designed to temporary remove the low-quality samples from the model training process.

MAE and MSE are the evaluation metrics used to evaluate the designed SiCNN model. These results will be presented and discussed in detail in chapter 4.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

This chapter focuses on the results and discussion from this study. Section 4.2 focuses on the training and testing results obtained from the SiCNN model. The results for the SiCNN model enhanced with the fast lane (FL) method and the sample selective (SS) method are discussed in section 4.3. This section also covers the benchmark the SiCNN model with the existing research works. Section 4.4 presented the cross-dataset performance using the proposed designed model and the training strategy. Section 4.5 discusses the implementation and deployment of the SiCNN model on the Edge platform. The evaluation of the inference speed is also presented in this chapter. The last section summarises the results achieved in this chapter.

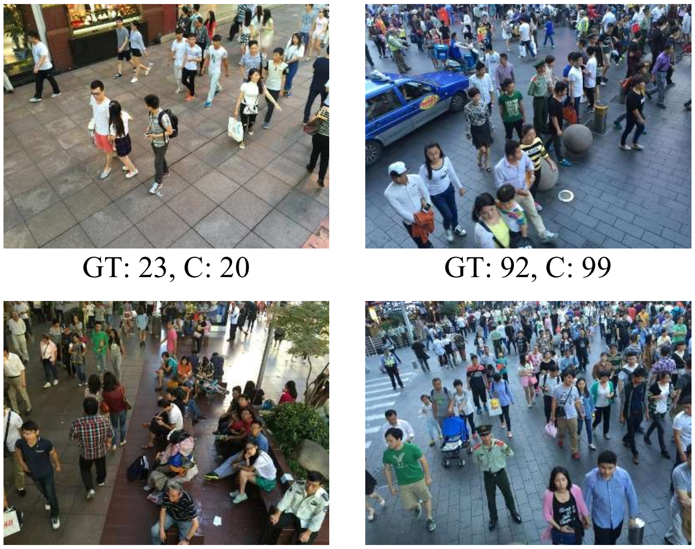
4.2 SiCNN Model Results and Analysis

The proposed designed SiCNN model was trained and evaluated with the ShanghaiTech part B dataset. As discussed in section 3.5, ShanghaiTech part B was chosen for validation in this research because the dataset contains several different scenes. This dataset fulfils the objective of analysing the performance of the model in scene invariant crowd prediction. The model was trained with the training methodology presented in section 3.3. The primary evaluations were conducted on an Ubuntu 18.04 LTS (Long Term Support) computer using an AMD Ryzen Threadripper with 16 cores (2.0 GHz per core) and GeForce RTX 2080 Ti with 4352 CUDA core. The SiCNN model is implemented in Python 3.7.5 using PyTorch v1.7.1 libraries. PyTorch is a machine learning framework based on the open-source Torch library that excel in creating deep neural networks. It is one of the preferred platforms for deep learning research. Five-fold cross-validation is deployed to evaluate the model using the MAE and MSE evaluation metrics (discussed in section 3.4). The variations in the MAE and MSE scores were recorded, beginning with high values at the initial epochs and eventually reaching convergence at 200 epochs. The MAE and MSE scores were obtained at each iteration of the five-fold cross-validation, yielding a total of five sets of scores. The final accuracy score of the SiCNN model was determined by averaging these scores. The results were verified and given in Table 4.1.

Table 4.1: MAE and MSE results of the SiCNN model.

Five-fold cross-validation	MAE	MSE
1st iteration	14.6	22.3
2nd iteration	13.6	21.3
3rd iteration	13.5	20.8
4th iteration	13.4	20.4
5th iteration	13.1	20.9
Average	**13.6**	**21.1**

The SiCNN model crowd prediction accuracy is evaluated and achieved the MAE of 13.6 and MSE of 21.1. The scores were calculated by taking the average of the results from each iteration in the five-fold cross-validation. This is the baseline results for the SiCNN model. The recorded MAE score from each iteration is between 13.1 and 14.6, while the recorded MSE score is between 20.8 and 22.3. The results show that the prediction error of the model is approximately ±13 counts from ground truth. As an example, if the actual number of people in a crowd image is 100, the model's prediction of the crowd size would typically fall within a range of 90 to 110 people. Some of the results are illustrated in Figure 4.1.



GT: 23, C: 20

GT: 92, C: 99

GT: 54, C: 48

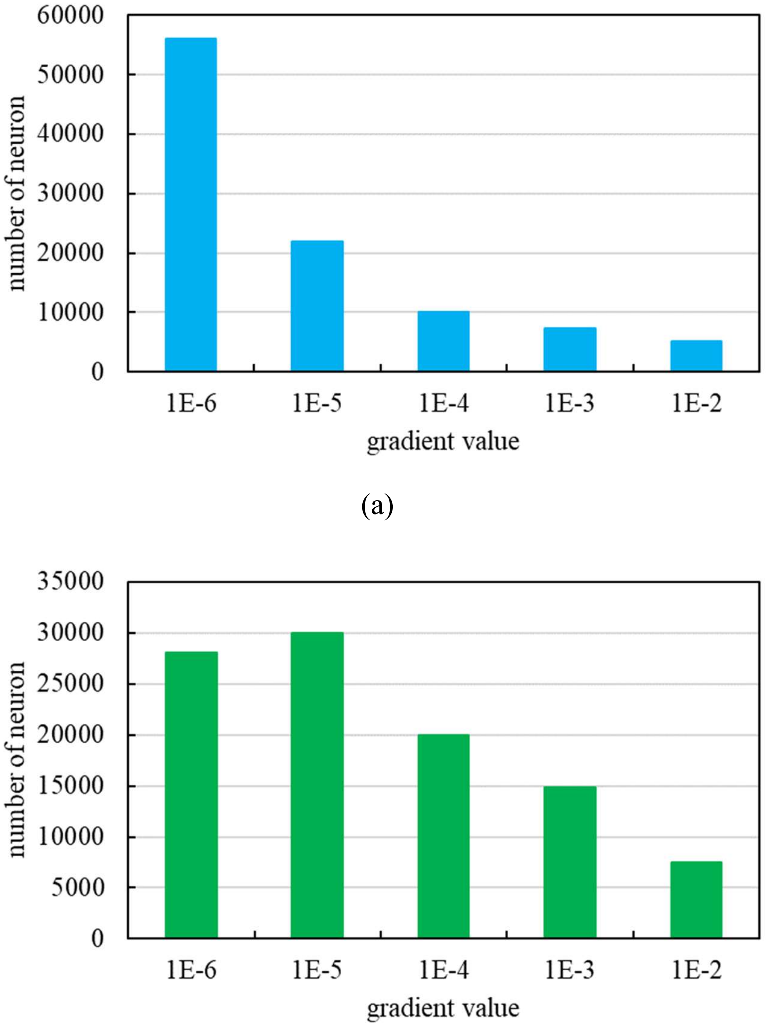
GT:170, C: 181

Figure 4.1 : Some results on the ShanghaiTech part B dataset. GT is the ground truth of the image, and C is the predicted count by the SiCNN model.

4.3 SiCNN Model using Fast Lane and Sample Selective Methods Results and Analysis

Section 3.6 and 3.7 addressed and examined the network degradation and the issue with the dataset participation. The investigation revealed that the network performance was impacted by the vanishing gradients and low-quality samples. Therefore, in chapter 3, the fast lane method (FL) and the sample selective method (SS) were discussed to improve network performance.

To assess the efficacy of the fast lane technique in enhancing network gradient, gradients from one of the convolutional layers were recorded prior to and after implementing the method. The changes are illustrated in Figure 4.2.



(a)

(b)

Figure 4.2 : The results of fast lane method deployment to the SiCNN model. (a) before deployment, (b) after deployment.

From the Figure 4.2 (a), the distribution of the neurons contained small value of gradient are mainly gathered at the exponent of -6. There were approximately 50,000 neurons with gradients exponentiating to -6, 20,000 neurons with gradients exponentiating to -5, 10,000 neurons with gradients exponentiating to -4, 7,000 neurons with gradients exponentiating to -3, and the remainder of the neurons had gradients exponentiating to -2. The observation indicated that when the gradient is extremely small, the network's weights are barely updated to new values. Upon deployment of the fast lane method to the model, Figure 4.2 (b) displays a slight shift in the gradients away from exponent -6, with the gradients now primarily concentrated at exponent -5. The results demonstrate that the deployment of the fast lane method can improve the gradient values to a healthier (slightly larger) range, which allows the network to converge more effectively. The SiCNN model was retrained using the ShanghaiTech part B dataset and the same training methodology (as described in section 3.3) with the implementation of the fast lane method, and its performance was re-evaluated using the same evaluation metrics (outlined in section 3.4). This model is named as SiCNN + FL. The new results are presented in Table 4.2.

In order to reduce the participation of low-quality samples in the model training, a sample selective algorithm is introduced and discussed in section 3.7. The algorithm is designed to identify non-essential and outlier samples, which are then temporarily excluded from the model training. The model is retrained using the same methodology and is now referred to as SiCNN + FL + SS. To clearly showcase the performance of the algorithm, the training curves for the

SiCNN network, SiCNN + FL network, and the SiCNN + FL + SS network are illustrated in Figure 4.3.

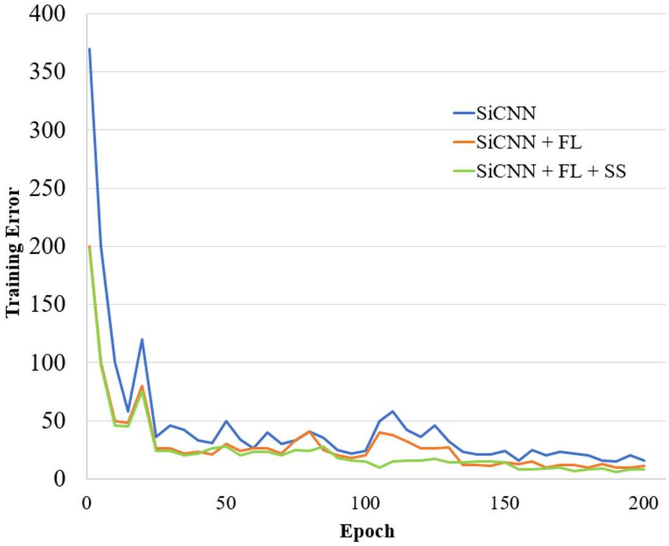


Figure 4.3 : The contrast curves of the training error between the SiCNN model, SiCNN + FL model and SiCNN + FL + SS model. FL: Fast Lane. SS: Sample Selective

As illustrated in the figure, it can be easily observed that the curve corresponding to the SiCNN network is more oscillating and its amplitude is larger in the early training stage. With the deployment of fast lane (FL) method, its amplitude is reduced and the overall training error is improved compared with the SiCNN network. The introduction of sample selective (SS) method where the curve corresponding to the SiCNN + FL + SS network indicate the method improves the stability and facilitates the network convergence. The performance of the proposed model is illustrated in Figure 4.4. Besides, the methods are compared with the existing research works and the results are given in Table 4.2. Some of the results, using the same image index numbers as in Figure 4.1, are displayed in Figure 4.5.

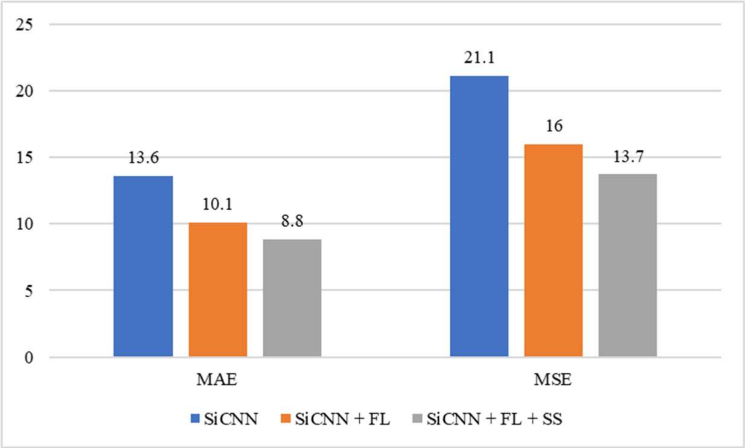
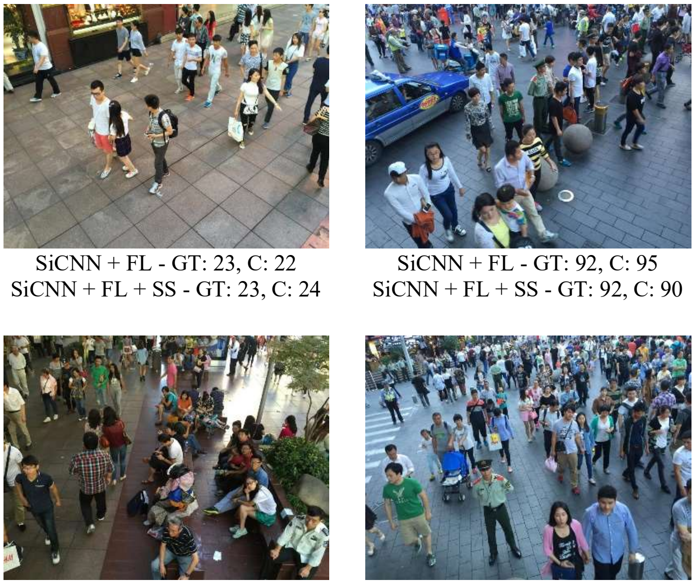


Figure 4.4 : Performance evaluation on ShanghaiTech Part B dataset. The MAE and MSE are clearly decreased from SiCNN to SiCNN + FL + SS model.

Table 4.2: Comparison of performance (MAE and MSE) of the proposed approaches with the state-of-art CNN-based methods.

Method	MAE	MSE
Switching-CNN (Sam, Surya and Babu, 2017)	21.6	33.4
BSAD (Huang et al., 2018)	20.2	35.6
Cascaded-MTL (Sindagi and Patel, 2017)	20.0	31.1
Multi-Scale CNN (Zeng et al., 2018)	17.7	30.2
DRSAN (Liu et al., 2018a)	11.1	18.2
Fully-CNN (Liu et al., 2018b)	10.1	18.8
LSC-CNN (Sam et al., 2021)	8.1	12.7
SiCNN (ours)	13.6	21.1
SiCNN + FL (ours)	10.1	16.0
SiCNN + FL + SS (ours)	8.8	13.7



SiCNN + FL - GT: 23, C: 22
SiCNN + FL + SS - GT: 23, C: 24

SiCNN + FL - GT: 92, C: 95
SiCNN + FL + SS - GT: 92, C: 90

SiCNN + FL - GT: 54, C: 52
SiCNN + FL + SS - GT: 54, C: 54

SiCNN + FL - GT: 170, C: 165
SiCNN + FL + SS - GT: 170, C: 173

Figure 4.5 : Some results on the ShanghaiTech part B dataset. GT is the ground truth of the image, and C is the predicted count from the model respectively.

As indicated in the table, the SiCNN model produced baseline results of a MAE of 13.6 and a MSE of 21.1. The errors obtained are lower compared to the results from previous studies by (Sam, Surya and Babu, 2017; Sindagi and Patel, 2017; Huang et al., 2018; Zeng et al., 2018), who employed a multi-column network design approach. The implementation of the fast lane connection technique leads to a reduction in the crowd estimation error, resulting in an improvement of 25.7% in MAE to 10.1 and 24.2% in MSE to 16.0. The reduction in error demonstrates that the fast lane connection method effectively addresses the problem of vanishing gradients, which was a contributing factor to the degradation of the network.

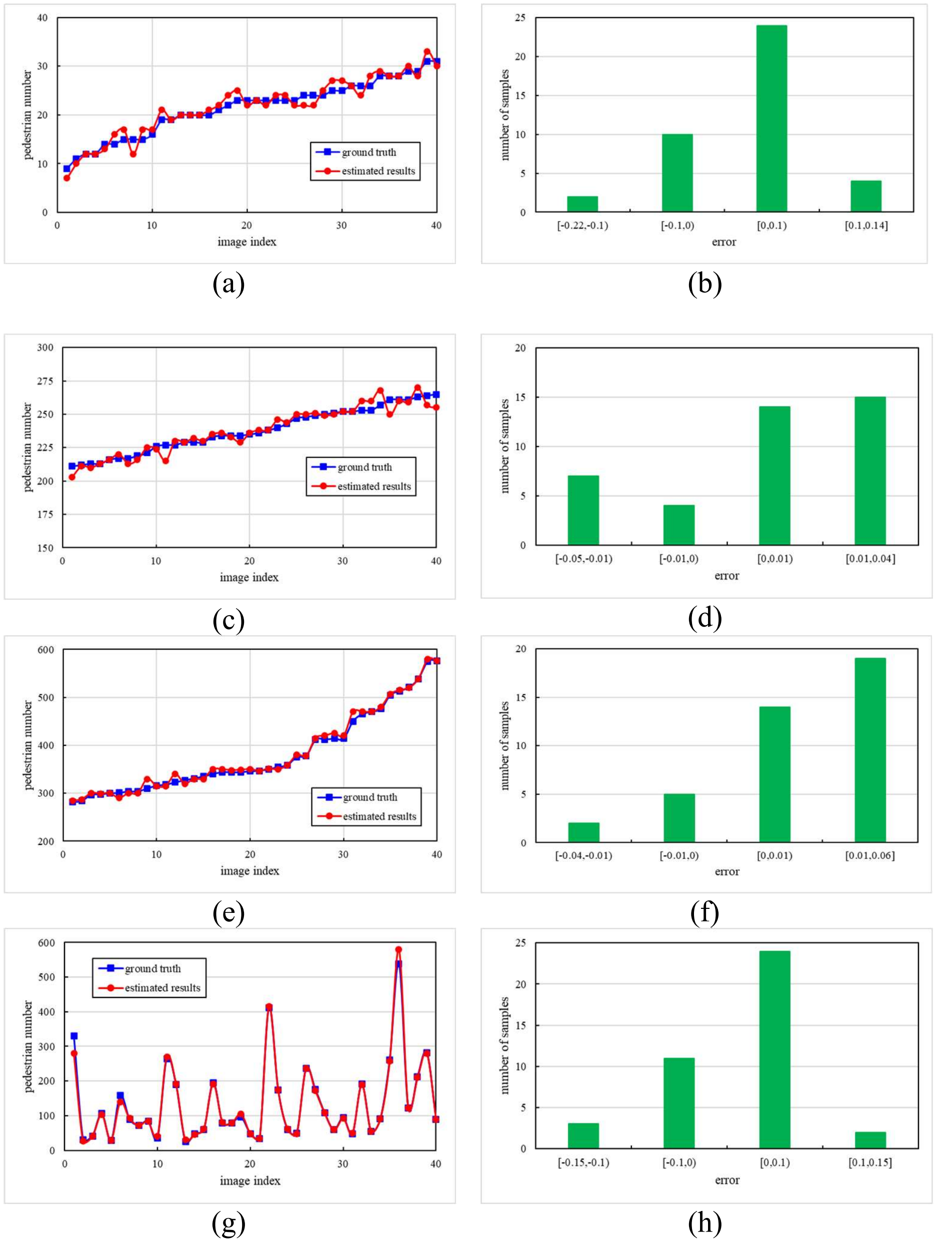
The introduction of the sample selective algorithm led to a further decrease in the MAE and MSE scores, with the scores now being 8.8 (a 12.9%

improvement) and 13.7 (a 14.4% improvement), respectively. Training with the removal of low-quality samples shows significant advantage in error reduction. The results were compared with research using the single-column design (Liu et al., 2018a; 2018b) and showed better accuracy. The SiCNN + FL + SS model exhibits slightly weak results compared to the LSC-CNN model, with a difference of 0.2-1.0 in the MAE and MSE scores between the two models. Their work uses a dense detection framework to detect heads of human. However, heavy annotation of the human head must be done for each new scene, leading to human error and a large amount of time spent on verification. In total, the SiCNN + FL + SS model showed a 35% and 39% reduction in MAE and MSE respectively, compared to the baseline SiCNN model.

To analyse the concrete performance of the proposed model using the ShanghaiTech part B dataset, comparison between the estimated results and the ground truth on several images selected from the dataset is conducted. To provide a clearer and more intuitive comparison, 40 images from the dataset have been selected and divided into four categories based on crowd density. These categories are low density, medium density, high density and dynamic density. As described in section 3.4, the evaluation metrics MAE and MSE only indicate the prediction error of the model but not the predictive behaviour of the model. To thoroughly evaluate the performance of the model in estimating crowd density, an error rate distribution analysis is performed on the four crowd density categories. The formula for the error rate distribution is defined as follows:

$$E_i = \frac{y_i' - y_i}{y_i} \tag{4.1}$$

where E_i represents the error rate of the i^{th} image, y_i' represents the estimated results of the i^{th} image and y_i represents the ground truth of the i^{th} image. Analysis of the error rate distribution can measure how often the model underestimates or overestimates the size of the crowd. The comparison between the estimated results and the ground truth based on the four categories of crowd density are shown in Figure 4.6 (a), Figure 4.6 (c), Figure 4.6 (e) and Figure 4.6 (g) respectively. The error rate distribution for each group is depicted in Figure 4.6 (b), Figure 4.6 (d), Figure 4.6 (f) and Figure 4.6 (h) in that order. The crowd density categories are ordered from low density to medium density, then to high density and finally to dynamic density.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 4.6 : Performance of the SiCNN network on Shanghaitech part B dataset. (a), (c), (e) and (g) are the comparison of the estimated results and ground truth for crowd density of low, medium, high and dynamic respectively. (b), (d), (f) and (h) are the error rate distribution for crowd density of low, medium, high and dynamic respectively.

The comparison figures indicate that the estimated results are mostly aligned with the ground truth. The images with a crowd size less than 30 pedestrians are classified as low-density, while those with a crowd size ranging between 200 and 270 are categorized as medium-density. The images with a crowd size greater than 300 pedestrians are considered high-density. The dynamic density category contains images randomly selected from the ShanghaiTech part B dataset. The results show that there is some fluctuation in the crowd estimation as the crowd size increases. However, the model performs fairly well in predicting the dynamic crowd size.

From the distribution, it can be seen that the error rate in the four categories of crowd density lies in the interval of [-0.22, 0.14], [-0.05, 0.04], [-0.04, 0.06] and [-0.15, 0.15] respectively. All the distribution are mainly centralised in the interval of [0, 0.1]. In fact, it can be deduced from this analysis that the model slightly overestimated the size of the crowd, given the difficult characteristics of the ShanghaiTech part B dataset. Nevertheless, the proposed designed model and methods have a preferable performance on this dataset.

4.4 Cross-dataset Evaluation

To show that the proposed model has better generalise to unseen scenes, cross-dataset evaluation is conducted. Instead of evaluating the quality of the model solely based on one dataset, cross-dataset evaluation is capable to evaluate the model performance from a different angle. Model with better generalisation enables the ability to adapt properly to new, previously unseen data rather than always retrain the model for a new scene which is often time consuming. In this experiment, the UCSD dataset and the Mall dataset are treated as unseen scenes for the SiCNN + FL + SS model. For cross-dataset evaluation, 40 samples are selected from each testing set and are evaluated using the proposed model. The evaluation is carried out using the MAE and MSE metrics.

The experimental results are given in Table 4.3 and illustrated in Figure 4.7. The cross-dataset evaluation is extended to benchmark with the state-of-art approaches. The benchmark results are given in Table 4.4 and Table 4.5 for UCSD dataset and Mall dataset respectively. Some of the results are illustrated in Figure 4.8. The error distribution was analysed again to evaluate the results, and the graphs are presented in Figure 4.9. Some of the comparison examples are illustrated in Figure 4.10.

Table 4.3: Generalisation to unseen datasets. "S_B", "U" and "M" are denoted as ShanghaiTech part B, UCSD and Mall. In "S_B→U" and "S_B→M" indicate ShanghaiTech part B is used for training, then test on UCSD and Mall.

	S_B→U		S_B→M	
Method	MAE	MSE	MAE	MSE
SiCNN	4.13	5.23	3.22	3.97
SiCNN + FL	3.79	4.84	2.75	3.66
SiCNN + FL + SS	3.29	4.21	2.21	3.09

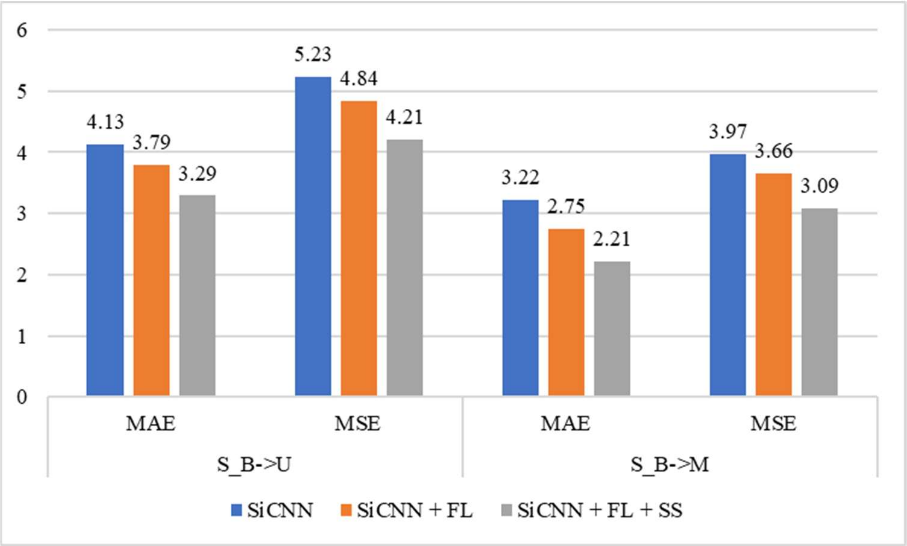


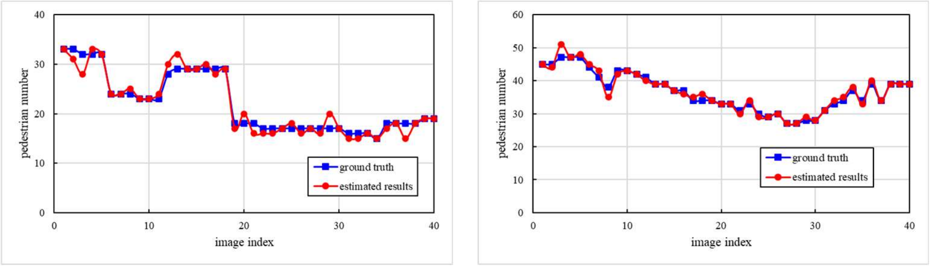
Figure 4.7 : Cross-dataset evaluation results of the proposed model.

Table 4.4: Comparison of performance (MAE and MSE) of the cross-dataset evaluation results with the state-of-art methods tested in UCSD dataset. FL: Fast Lane. SS: Sample Selective.

Method	MAE	MSE
Weighted V-LAD (Sheng et al., 2018)	2.86	3.61
Switching-CNN (Sam, Surya and Babu, 2017)	1.62	2.10
Cross-scene CC (Zhang et al., 2015)	1.60	3.31
DA-NET (Zou et al., 2018)	1.03	1.31
SiCNN (ours)	4.13	5.23
SiCNN + FL (ours)	3.79	4.84
SiCNN + FL + SS (ours)	3.29	4.21

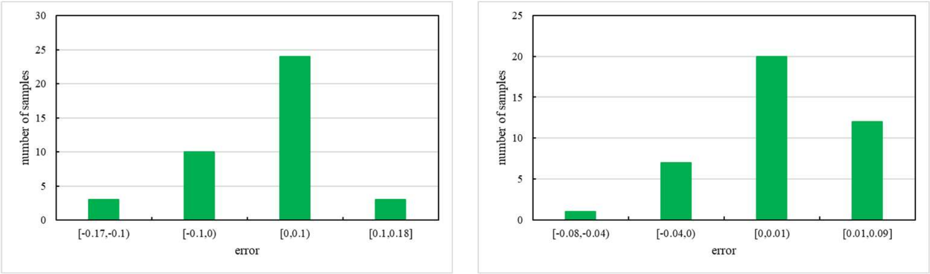
Table 4.5: Comparison of performance (MAE and MSE) of the cross-dataset evaluation results with the state-of-art methods tested in Mall dataset. FL: Fast Lane. SS: Sample Selective.

Method	MAE	MSE
MoCNN (Kumagai, Hotta and Kurita, 2018)	2.75	3.66
DRSAN (Liu et al., 2018a)	1.72	2.10
E3DNet (Zou et al., 2020)	1.64	2.13
SAANet (Hossain et al., 2019)	1.28	1.68
SiCNN (ours)	3.22	3.97
SiCNN + FL (ours)	2.75	3.66
SiCNN + FL + SS (ours)	2.21	3.09



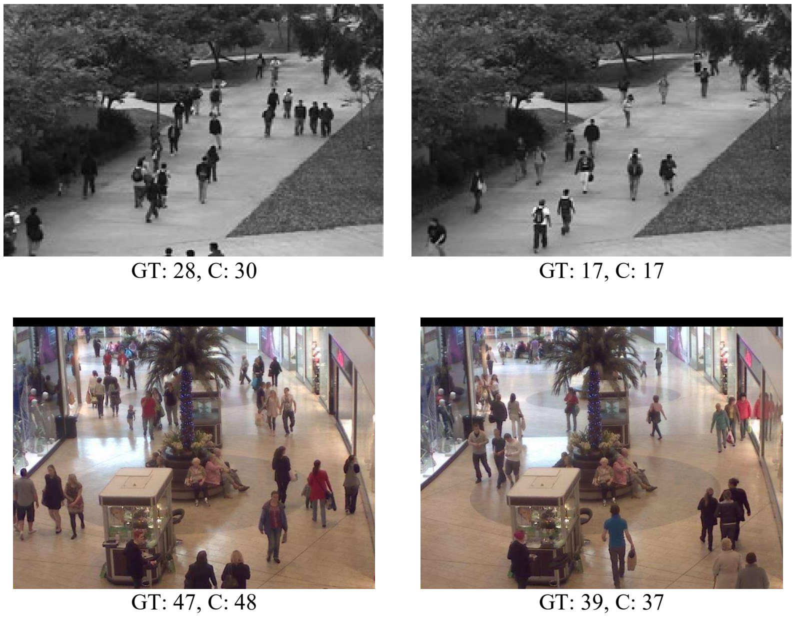
(a) (b)

Figure 4.8 : Cross-dataset evaluation results of the SiCNN + FL + SS network on unseen dataset. (a) UCSD dataset (b) Mall dataset.



(a) (b)

Figure 4.9 : Error distribution analysis of the SiCNN + FL + SS network on unseen dataset. (a) UCSD dataset (b) Mall dataset.



GT: 28, C: 30

GT: 17, C: 17

GT: 47, C: 48

GT: 39, C: 37

Figure 4.10 : Some results on the UCSD dataset (top) and Mall dataset (bottom).

The SiCNN model showed results of MAE 4.13 and MSE 5.23 when evaluated on the UCSC dataset and MAE 3.22 and MSE 3.97 when evaluated on the Mall dataset, as indicated in the table. The SiCNN + FL + SS model further reduced the errors to 3.29 for MAE and 4.21 for MSE when evaluated using the UCSC dataset, and to 2.21 for MAE and 3.09 for MSE when evaluated using the Mall dataset. To summarize, the SiCNN + FL + SS model resulted in a 20.3% decrease in MAE and a 19.5% decrease in MSE for the UCSC dataset, and a 31.4% decrease in MAE and a 22.2% decrease in MSE for the Mall dataset. The results further demonstrated the proposed designed algorithms have advantages in improving the network's performance. The estimated results shown in Figure 4.8 are mostly in line with the ground truth. While there are a few spikes present, the overall curve is relatively smooth. Based on the error distribution analysis, the model's crowd prediction behaviour is consistent with previous results presented in Figure 4.6, with the model tending to overestimate crowd size in most cases.

Benchmarking is a process to measure the quality and performance of the network with the existing work. It can determine the gap between the results and the existing state-of-art research works. To analyse the cross-dataset evaluation performance, comparison between the results and the existing research works was conducted. From the comparison given in Table 4.4 and Table 4.5, the proposed model could not outperform the current state-of-art methods. Their methods have 13% to 40% better performance compared with the SiCNN + FL + SS model. However, their research works are conducted using the training-testing protocol. It is envisaged that their models generally cannot achieve good accuracy when testing on unseen scenes. Thus, their methods did not have better generalisation to previously unseen data. To adapt to a new scene, extra effort is needed to retrain the model for any new dataset which is time intensive.

As of this point, the proposed designed model had been evaluated with the evaluation metrics and cross-dataset evaluation. The results were studied and discussed comprehensively in the previous section. Last but not least, this study conducted an evaluation to determine the minimum level of accuracy that is acceptable to the end user. To conduct the analysis, this research work utilised the mean relative error (MRE). The MRE was defined how large the error is relative to the actual value (ground truth). The definition of MRE is given as follows:

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i' - y_i}{y_i}\right| \times 100\% \tag{4.2}$$

where N is the number of test data, y_i is the ground truth and y_i' is the estimated results corresponding to the i^{th} image. According to an internet study conducted by Regazzoni (Regazzoni, Tesei and Murino, 1993), “The end users accept a mean error of 20% with respect to the real number of people present in a scene”. Due to the study was carried out few decades ago, this research work hypothesizes that nowadays with the advancement of technology and better algorithms, the end users can accept a mean error of 10% with respect to real number of people appear in a scene. A lower MRE value indicates a higher estimation precision. The same 40 samples selected from Shanghaitech part B dataset, UCSD dataset and Mall dataset were used to assess the level of acceptance by end-user. The proposed designed model achieved an average MRE of 7.8%. The achievement meets the minimum accuracy requirements of the system operator (end user).

4.5 Inference Speed Analysis

Developing a low-latency crowd counting model that can be leveraged in real time is another interesting question and rarely addressed by researchers. There is a contradiction between the inference speed and the accuracy. This is known by speed-accuracy trade-off (Huang et al., 2017). The hypothetical speed-accuracy trade-off curves is illustrated in Figure 4.11.

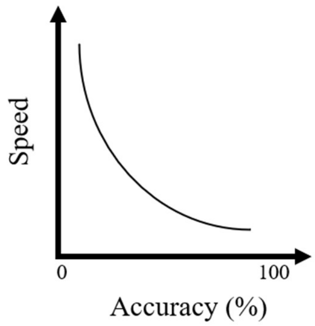


Figure 4.11 : Hypothetical speed-accuracy trade-off curves of the network.

According to the figure, a model with high accuracy may have a slow inference speed. While most existing methods have impressive accuracy, they tend to use deep model structures with a large number of layers. As the number of layers increases, more neurons must be processed, leading to a decrease in the inference speed. Since most research works did not evaluate the inference speed, it is speculated that their results do not meet the level of performance that would be acceptable to end users in terms of execution speed.

It is important to assess the inference speed of the proposed model so that it meets the standards of acceptance by the end user. With the invention of an affordable and powerful edge embedded platform, many deep learning algorithms are implemented on the edge side. The benefit of processing at the edge platform is to reduce the strain on network bandwidth by avoiding the transmission of large amounts of images to the server, which can cause lag issues. By performing crowd prediction at the edge, only the results are sent to the central server. Therefore, to better evaluate the inference speed, the proposed model is tested on the NVIDIA Tx2 board and the Intel Up board, both of which are specifically designed for edge processing.

The specification of the NVIDIA Tx2 board consists of a quad-core 2GHz ARM CPU, 8GB of RAM and 256 CUDA cores. The test was implemented using the same computer setting mentioned in section 4.2, which consists of an Ubuntu operating system, Python 3.7.5 and PyTorch v1.7.1 libraries. The inference of the model was configured to be executed using the CUDA cores where the memory allocation and the calculation in the CUDA cores being managed automatically by the PyTorch framework. Despite the limited resources of the embedded board, the proposed model is capable of performing crowd estimation at an average inference speed of 2.51 seconds.

On the other hand, the test on the Intel Up board was conducted using a quad-core 1.44GHz Atom CPU, 4GB of RAM and an integrated 500MHz Intel GPU. The testing was carried out using the same operating system and library settings as in the NVIDIA Tx2 board, but with the added use of the OpenVINO toolkit. The toolkit consists of two components: the model optimiser (MO) a.k.a. the trained model and the inference engine (IE). The proposed designed model was configured to be optimised by the MO and generates an IE for optimal performance when running in the Intel GPU. A user application is programmed to interact with the IE to obtain the results. The OpenVINO architecture is illustrated in Figure 4.12. The proposed model can achieve an average inference speed of 3.43 seconds, which is slightly slower than the NVIDIA Tx2 board. This is because the CUDA core excels at matrix multiplication, and deep learning models heavily rely on this process. The results reiterate the suitability of the proposed model for practical applications. To ensure thoroughness, the proposed model's inference speed is measured using the computing specifications described in section 4.2. The model achieved an average inference speed of 31.7ms.

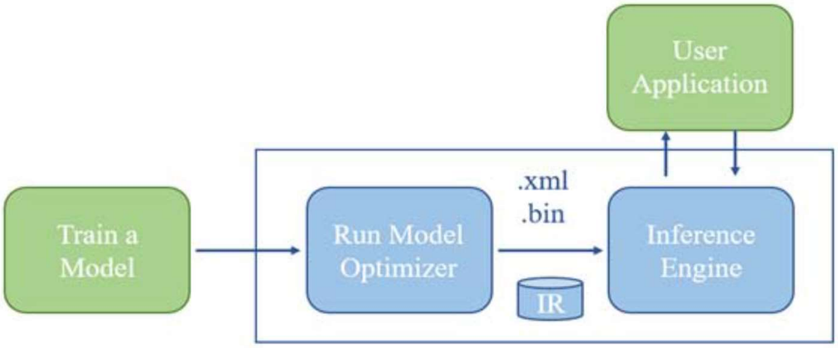


Figure 4.12 : OpenVINO architecture used in this research work. (Castro-Zunti, Yépez and Ko, 2020)

4.6 Summary

In this chapter, the proposed designed SiCNN model, labeled as the baseline model is evaluated using the evaluation metrics (MAE and MSE) through five-fold cross validation. The Shanghaitech part B dataset was used as a training and testing set. The results were recorded at 13.6 for MAE and 21.1 for MSE. Some of the results are shown in Figure 4.1. To address the issue of network degradation causing vanishing gradients, the baseline model was further improved by introducing the fast lane method, which facilitates the transfer of larger gradients from the early layers to the later layers. The verification of the method is given in Figure 4.2. The Fast Lane method has been proven to enhance the gradients to a more desirable value instead of a very small value. The baseline model was retrained to obtain a new set of results, leading to the creation of a new model named SiCNN + FL. The SiCNN + FL model achieved a MAE of 10.1 and MSE of 16.0, resulting in a 25.7% improvement in MAE and a 24.2% improvement in MSE.

Further investigation of the dataset participation was conducted. Sample selective method was introduced to weight down the non-essential sample and outlier sample. These samples are temporary “sleep” for a few epochs during the training and “wake” to rejoin the training process. The sample selective method was successful in reducing the error to 8.8 MAE (an improvement of 12.9%) and 13.7 MSE (an improvement of 14.4%) compared to the SiCNN + FL model. The new model is referred to as the SiCNN + FL + SS model. In total, the MAE and MSE of the SiCNN + FL + SS model improved by 35% and 39% respectively from the baseline SiCNN model. Lastly, the model was benchmarked with the current state-of-art research works.

Cross-dataset evaluation is conducted to assess the scene invariant performance. The final model SiCNN + FL + SS was trained using the Shanghaitech part B dataset while the UCSD dataset and Mall dataset were treated as unseen data. The evaluation of the model's scene invariant performance recorded MAE values of 3.29 and 2.21 for the UCSD and Mall datasets respectively, with corresponding MSE values of 4.21 and 3.09. These results were compared with those from previous works.

Finally, the final model was deployed on two edge computing platforms, the NVIDIA Tx2 board and Intel Up board. The inference speed was measured and recorded to be 2.51 seconds on the NVIDIA Tx2 board and 3.43 seconds on the Intel Up board. The NVIDIA Tx2 board demonstrated better performance in terms of speed due to its strong ability in matrix multiplication, which is crucial for deep learning models.

In conclusion, the results obtained were thoroughly analysed, verified, and discussed in detail in the respective sections of this chapter.

CHAPTER 5

DISCUSSION AND CONCLUSION

5.1 Introduction

In the introduction to this thesis the problems associated with crowd counting were discussed in detail. The discussions pointed out that although significant improvements have been made in estimating the crowd density from a complex scene, the problem of the crowd counting model that can perform in different scene remains a concern. The existing models perform well when testing with the dataset that is also used for training. Their accuracy drops drastically when models are applied to unseen datasets or unseen scene. Therefore, this thesis argues that this justifies putting efforts into research in scene invariant crowd counting.

5.2 Traditional Crowd Counting Method Problem

The discussions in section 1.1 highlighted the limitation of traditional crowd counting approaches. In the early stages of research, detection and regression approaches utilising hand-crafted features were commonly used for crowd prediction. However, crowd scenes bring numerous challenges such as occlusions, objects that are both static and dynamic, non-uniform distributions of people, and uneven illumination, which limit the effectiveness of these approaches. As a result, the advent of CNNs in addressing computer vision challenges has led to the implementation of CNN-based crowd counting in this field of research, replacing the previously used hand-crafted feature detection and regression methods which had limitations in handling occlusions, static and dynamic objects, non-uniform distribution of people, and non-uniform illumination in a crowd scene. The main purpose of this thesis was to analyse the capability of CNN methods and then design a new model suitable for uncomplicated implementation in crowd counting applications.

5.3 Density Map Problem

Another problem that is highlighted in the section 1.1 is the crowd estimation based on density map. The difficulties lie in the creation of density maps and making predictions based on them. Prior to training a model to predict the density map, it is necessary to generate the ground truth density map from the dataset through the process of density map generation using a Gaussian kernel. However, the bandwidth parameters of the kernel are often selected manually and can be dependent on the specific dataset, meaning that the same parameters may not perform optimally on other datasets. It is a time-consuming process to manually select the parameters for each new dataset.

Furthermore, the max-pooling layers employed in the network and the Euclidean loss used for optimisation have been recognised to be problematic, leading to a decrease in accuracy due to the resulting down-sampled density map. Therefore, this thesis chose to use global regression for crowd prediction, where the total crowd size is estimated directly from the image without relying on the density map. This approach avoids the issues associated with training the model on low-quality density maps.

5.4 Specific Scene Crowd Counting Problem

Section 2.4's discussions emphasised the issue with the crowd counting model's limitation in only performing optimally in specific environments. The main goal of a deep learning model is able to achieve good generalisation ability. Current methods typically adopt the training-testing protocol within a single dataset and their accuracy declines when applied to unseen datasets. The majority of existing methods require retraining on a new scene, which is not practical in real-world scenarios as it would be costly to obtain annotations for every new scene. Thus, the main motivation for this thesis was to develop a new model that generally only needs to be trained once and can perform crowd counting in diverse, previously unseen environments with a reasonable degree of accuracy.

5.5 Designing a Scene Invariant Convolutional Neural Network Model

This thesis has investigated and developed a Scene Invariant Convolutional Neural Network (SiCNN) model, driven by the benefits of a CNN model capable of performing crowd counting in diverse scenarios. The SiCNN model is designed to carry out cross-scene crowd counting without the requirement of re-training the model specifically for each new scene. The methodology for designing the SiCNN model is thoroughly discussed in Chapter 3.

The architecture of the SiCNN model is designed with 13 convolutional layers and 6 fully connected layers, inspired by the superior generalisation and success of the VGG-16 architecture. The selection of a 3x3 filter size as the kernel or filter size in this study brings advantages in terms of reduced trainable parameters while still preserving the image details. Batch Norm technique is applied as pre-activation to normalise the neurons' output before the activation function to avoid overfitting. To effectively highlights the image features and facilitate flexible crowd size estimation, as reported in section 3.2, the ReLU activation function is utilised in convolutional layers and ELU activation function is employed in fully connected layers. As reported in section 2.7, Kaiming weight initialisation strategy is chosen to initialise the network weights because it works well with the chosen activation functions.

Finally, the SiCNN model is trained using the ShanghaiTech part B dataset. As reported in section 4.2, ShanghaiTech part B dataset contains multiple different scenes that compliance the objective to support the proposed designed model learn to perform crowd counting across different scenes. The evaluation results revealed that the SiCNN model achieved a lower MAE and MSE score in comparison to the multi-column model, which is a commonly utilised method for density map prediction. The evaluation results also clearly indicate that further error reduction could be accomplished by addressing the vanishing gradient problem in the network and the participation of low-quality samples in the model training. These problems were reported in section 4.3.

5.6 Solving the Vanishing Gradient Problem

The network's degradation problem that highlighted in section 3.6 is the vanishing gradient. This problem results in the difficulty of updating the network's weight to new values during model training when the gradient is very small. This leads to the network convergence reduction. The investigation results are reported in Figure 3.5 and Figure 3.6 where the very small gradient are found in the convolutional layers and fully connected layers. Therefore, the fast lane method is proposed to address this problem where it is designed as a "bridge" to connect the output of a layer into the layer about few hops away. The analysis reported in section 4.3 show that the fast lane method enables the learned features from the earlier layers to propagate to the later layer resulting in the gradient values are shifted to a healthy level that adequate to update the network's weights. Therefore, this led to improved model convergence and a reduction in prediction error, resulting in the creation of the SiCNN + FL model. The details of this approach are documented in section 3.6. The results of implementing the fast lane method are presented in Table 4.2.

5.7 Solving the Low-quality Samples Problem

Another concern in the model training process was the inclusion of low-quality samples. The analysis reported in section 3.7 revealed the presence of samples that did not contribute to the learning of the model. These samples are either accurately predicted by the model early on or their predicted count consistently deviates significantly from the ground truth. As a result, a sample selection algorithm was developed to identify non-essential and outlier samples based on the prediction error using threshold values. These samples are temporarily removed from the training process, as indicated by the activation time shown in Figure 3.10. By down-weighting the low-quality samples, the model accuracy was improved and the prediction error was further reduced. This resulting model is referred to as SiCNN + FL + SS. The detail of this method is documented in section 3.7. The result of sample selective algorithm is reported in table 4.2.

5.8 Scene Invariant Performance

The scene invariant performance of the proposed designed model (SiCNN + FL + SS) was evaluated and studied and the results are reported in section 4.4. This model was trained with the ShanghaiTech part B dataset where it is rich with different crowd scene. Two additional datasets, considered as unseen or untrained scenes, were then used to evaluate the proposed model. The scene-invariant performance was evaluated using the MAE and MSE metrics. The evaluation results of the SiCNN + FL + SS model show that it falls within an acceptable range for end-users. The evaluation results were compared with the state-of-art methods reported in section 4.4. Although the proposed designed model could not outperform them, it is envisaged that their models only work well in single dataset that used to train the respective model and have possibility could not perform well in another dataset.

To counteract the drawback of the MAE and MSE metrics outlined in section 3.4, error rate distribution is employed. The metric assesses the accuracy of the model in predicting the crowd size, both in cases of overestimation and underestimation. Further information can be found in section 4.3. The SiCNN + FL + SS model performs well in crowd prediction regardless of the scene, with a tendency towards slightly overestimating the crowd size. The results are presented in section 4.3.

To achieve a certain level of comprehensiveness in this research, the inference speed of the designed model, which is seldomly mentioned in other research, was evaluated. Despite the intricate design, the SiCNN model delivers inference speed that is suitable for end-users. The novelty of this study lies in the combination of techniques applied to the SiCNN model. As a result, a deep learning model for crowd counting has been developed that can perform well in various crowd scenes with an efficient inference speed.

5.9 Contributions

This section describes the contributions made by this thesis. The contributions are classified into main contributions based on the work carried out in this thesis and contribution based on previous work. This classification is, of course, subjective and represents the author's opinion. The references in the parenthesis refer to the corresponding sections in the thesis.

5.9.1 Main Contributions from this Thesis

i. A convolutional neural network-based model for visual crowd counting was designed, developed and proven experimentally (section 4.2).

ii. An extended CNN-based crowd counting model for scene invariant crowd prediction was designed and tested (section 3.6 and section 4.4)

iii. A model optimisation strategy method for training has been proposed and tested (section 3.7 and section 4.3).

5.9.2 Other Contributions

i. Created an activation schedule for sample selective method activation during model training (section 3.7).

ii. Determined that the SiCNN + FL + SS model surpasses most of the CNN-based model designed with single-column, multi-column, density map prediction and crowd regression methods (section 4.3).

iii. Revealed from the experimental results that evaluation metrics have limitation in reflecting the quality of the model (section 3.4).

iv. Demonstrated from the measurement analysis that the proposed model meets the minimum accuracy requirement acceptable to the end-user (section 4.4).

v. Shown, with the aid of experimental work, the proposed model can attain an acceptable inference speed when executed on an edge platform with limited processing capabilities (section 4.5).

5.10 Limitation and Future Direction

One limitation of the current study is the lack of spatial information. The proposed designed model is capable to predict the total crowd size within a reasonable time frame. While the model results were studied and tested, the spatial information is absent. Spatial information can be valuable in determining the location of the crowd within an image. For example, the density map can identify the "hot spot" where the crowd gathers in a scene. This is particularly useful in crowd analysis applications. Despite its usefulness, generating density maps in real time is a complex task. The ability of the current model to predict the total crowd size in real time is expected to benefit the surveillance industry. The proposed designed model can be deployed in surveillance camera systems for real-time crowd prediction, reducing the strain on network bandwidth from transmitting images.

Another limitation that can affect the robustness of this study is the poor quality of camera input. Low quality camera input may introduce noise, artifacts or distortions in the images. This can lead to false patterns in the data, making it difficult for the model to correctly learn and generalise from the input. Moreover, blurriness in the camera input can obscure essential details, impacting the model's ability to highlight useful features that leads to disrupt the learning process and cause the model lack of crucial context to predict the crowd. Lastly, the robustness of a CNN model can be affected by the poor-quality camera that struggle with variations in lighting conditions. This leads to inconsistent image brightness and contract. The model may fail to adapt to such changes, affecting its generalisation capabilities.

Further studies should be conducted to minimize the crowd prediction error and enhance the model's performance in scene-invariant scenarios. The current results were not the best in terms of error compared to other research studies. A preliminary study on the use of transfer learning in crowd counting was recently conducted (Khalifa et al., 2022), however, it was only tested on the Mall dataset which is a relatively small evaluation scale. Further studies on leveraging a trained model through transfer learning to perform cross-scene crowd prediction should be conducted. The results obtained from such studies are crucial for improving the generalisation of the model.

Another direction of further studies is the implementation of multi-task framework. This technique not only account for crowd counting but also for task like crowd segmentation and crowd density classification. It is important to further furnish the designed model with deeper understanding of the crowd scene. Although multi-task framework can assist each other task to increase the overall accuracy of the network, the employability for real time application will be reduced due to the network complexity. It is a nascent area of research to strike the balance between the accuracy and the inference speed.

Finally, this study showed the deployment of the designed model on the edge platform, which achieved an acceptable inference speed. The current deployment used the default optimisation of the edge platform framework for arranging its resources during model inference. Further studies to create a customised framework for managing resources, such as allocating processing cores, to enhance the execution time of the model should be conducted.

REFERENCES

Andre, E., Brett, K., Roberto A, N., Justin, K., Susan M, S., Helen M, B. and Sebastian, T., 2019. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7).

Camgözlü, Y. and Kutlu, Y., 2020. Analysis of filter size effect in deep learning. arXiv preprint arXiv:2101.01115.
Castro-Zunti, R.D., Yépez, J. and Ko, S.B., 2020. License plate segmentation and recognition system using deep learning and OpenVINO. *IET Intelligent Transport Systems*, 14(2).

Chan, A.B., Liang, Z.S.J. and Vasconcelos, N., 2008. Privacy preserving crowd monitoring: Counting people without people models or tracking. In: *26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR*.

Chen, K., Loy, C.C., Gong, S. and Xiang, T., 2012. Feature mining for localised crowd counting. In: *BMVC 2012 - Electronic Proceedings of the British Machine Vision Conference 2012*.

Choi, H. and Lee, J., 2021. Efficient use of gpu memory for large-scale deep learning model training. *Applied Sciences (Switzerland)*, 11(21).

Clevert, D.A., Unterthiner, T. and Hochreiter, S., 2016. Fast and accurate deep network learning by exponential linear units (ELUs). In: *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*.

Dalal, N. and Triggs, B., 2005. Histograms of oriented gradients for human detection. In: *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*.

Datta Gupta, K., Sharma, D.K., Ahmed, S., Gupta, H., Gupta, D. and Hsu, C.H., 2023. A Novel Lightweight Deep Learning-Based Histopathological Image Classification Model for IoMT. *Neural Processing Letters*, 55(1).

Ding, B., Qian, H. and Zhou, J., 2018. Activation functions and their characteristics in deep neural networks. In: *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*.

Dubey, S.R., Singh, S.K. and Chaudhuri, B.B., 2022. Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing.
Duchi, J., Hazan, E. and Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12.

Fan, Z., Zhang, H., Zhang, Z., Lu, G., Zhang, Y. and Wang, Y., 2022. A survey of crowd counting and density estimation based on convolutional neural network. *Neurocomputing*, 472.

Felzenszwalb, P.F., Girshick, R.B., McAllester, D. and Ramanan, D., 2010. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9).

Fu, M., Xu, P., Li, X., Liu, Q., Ye, M. and Zhu, C., 2015. Fast crowd density estimation with convolutional neural networks. *Engineering Applications of Artificial Intelligence*, 43.

Fukushima, K., 1975. Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, 20(3–4).

Glorot, X. and Bengio, Y., 2010. Understanding the difficulty of training deep feedforward neural networks. In: *Journal of Machine Learning Research*.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Hossain, M.A., Hosseinzadeh, M., Chanda, O. and Wang, Y., 2019. Crowd counting using scale-aware attention networks. In: *Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019*.

Hu, Y., Chang, H., Nian, F., Wang, Y. and Li, T., 2016. Dense crowd counting from still images with convolutional neural networks. *Journal of Visual Communication and Image Representation*, 38.

Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. and Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*.

Huang, S., Li, X., Zhang, Z., Wu, F., Gao, S., Ji, R. and Han, J., 2018. Body structure aware deep crowd counting. *IEEE Transactions on Image Processing*, 27(3).

Ilyas, N., Shahzad, A. and Kim, K., 2020. *Convolutional-neural network-based image crowd counting: Review, categorization, analysis, and performance evaluation*. *Sensors (Switzerland)*.

Ioffe, S. and Szegedy, C., 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *32nd International Conference on Machine Learning, ICML 2015*.

Kaiming, H., Xiangyu, Z., Shaoqing, R. and Jian, S., 2018. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification Kaiming. *Biochemical and Biophysical Research Communications*, 498(1).

Khalifa, O.O., Albagul, A., Hashim, A.H.A., Hashim, N.A.M. and Zainuddin, K.N.S.W., 2022, May. Transfer Learning For Crowed Counting. In *2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and*

Techniques of Automatic Control and Computer Engineering (MI-STA) (pp. 248-253). IEEE.

Khan, A., Sohail, A., Zahoora, U. and Qureshi, A.S., 2020. A survey of the recent architectures of deep convolutional neural networks. *Artificial Intelligence Review*, 53(8).

Khan, M.A., Menouar, H. and Hamila, R., 2023. *Revisiting crowd counting: State-of-the-art, trends, and future perspectives*. *Image and Vision Computing*.

Keju, Z., Fuqiang, L. and Zhipeng, L., 2009. Counting pedeatrian in crowded subway scene. In: *Proceedings of the 2009 2nd International Congress on Image and Signal Processing, CISP'09*.

Khanday, O.M., Dadvandipour, S. and Lone, M.A., 2021. Effect of filter sizes on image classification in CNN: A case study on CFIR10 and fashion-MNIST datasets. *IAES International Journal of Artificial Intelligence*, 10(4).

Kingma, D.P. and Ba, J.L., 2015. Adam: A method for stochastic optimization. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Kong, D., Gray, D. and Tao, H., 2006. A viewpoint invariant approach for crowd counting. In: *Proceedings - International Conference on Pattern Recognition*.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6).

Kumagai, S., Hotta, K. and Kurita, T., 2018. Mixture of counting CNNs. *Machine Vision and Applications*, 29(7).

Li, W., Li, H., Wu, Q., Meng, F., Xu, L. and Ngan, K.N., 2020. HeadNet: An End-to-End Adaptive Relational Network for Head Detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(2).

Litjens, G., Sánchez, C.I., Timofeeva, N., Hermsen, M., Nagtegaal, I., Kovacs, I., Hulsbergen-Van De Kaa, C., Bult, P., van Ginneken, B. and van der Laak, J., 2016. Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. *Scientific Reports*, 6.

Liu, B. and Vasconcelos, N., 2015. Bayesian model adaptation for crowd counts. In Proceedings of the IEEE International Conference on Computer Vision (pp. 4175-4183).

Liu, L., Wang, H., Li, G., Ouyang, W. and Lin, L., 2018a. Crowd counting using deep recurrent spatial-aware network. In: *IJCAI International Joint Conference on Artificial Intelligence*.

Liu, M., Jiang, J., Guo, Z., Wang, Z. and Liu, Y., 2018b. Crowd counting with fully convolutional neural network. In: *Proceedings - International Conference on Image Processing, ICIP*.

Ma, Z., Hong, X., Wei, X., Qiu, Y. and Gong, Y., 2021. Towards A Universal Model for Cross-Dataset Crowd Counting. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Marsden, M., McGuinness, K., Little, S. and O'Connor, N.E., 2017a. Fully convolutional crowd counting on highly congested scenes. In: *VISIGRAPP 2017 - Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*.

Marsden, M., McGuinness, K., Little, S. and O'Connor, N.E., 2017b. Resnet-Crowd: A residual deep learning architecture for crowd counting, violent behaviour detection and crowd density level classification. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017*.

Mustapha, A., Mohamed, L. and Ali, K., 2020. An Overview of Gradient Descent Algorithm Optimization in Machine Learning: Application in the Ophthalmology Field. In: *Communications in Computer and Information Science*.

Narkhede, M. v., Bartakke, P.P. and Sutaone, M.S., 2022. A review on weight initialization strategies for neural networks. *Artificial Intelligence Review*, 55(1).

Nwankpa, C., Ijomah, W., Gachagan, A. and Marshall, S., 2018. Activation functions: Comparison of trends in practice and research for deep learning. arXiv preprint arXiv:1811.03378.

Oñoro-Rubio, D. and López-Sastre, R.J., 2016. Towards perspective-free object counting with deep learning. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Patwal, A., Diwakar, M., Tripathi, V. and Singh, P., 2023. Crowd counting analysis using deep learning: a critical review. *Procedia Computer Science*, 218.

Percannella, G., Conte, D., Foggia, P., Tufano, F. and Vento, M., 2010. A method for counting moving people in video surveillance videos. *Eurasip Journal on Advances in Signal Processing*, 2010.

Rahnemoonfar, M. and Sheppard, C., 2017. Deep count: Fruit counting based on deep simulated learning. *Sensors (Switzerland)*, 17(4).

Regazzoni, C.S., Tesei, A. and Murino, V., 1993. A real-time vision system for crowding monitoring. In: *IECON Proceedings (Industrial Electronics Conference)*.

Ryan, D.A., 2010. Crowd Monitoring Using Computer Vision. *Integrative Biology*, 2(2–3).

Sabzmeydani, P. and Mori, G., 2007. Detecting pedestrians by learning shapelet features. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Saleh, S.A.M., Suandi, S.A. and Ibrahim, H., 2015. *Recent survey on crowd density estimation and counting for visual surveillance*. *Engineering Applications of Artificial Intelligence*.

Sam, D.B., Peri, S.V., Sundararaman, M.N., Kamath, A. and Babu, R.V., 2021. Locate, size, and count: Accurately resolving people in dense crowds via detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(8).

Sam, D.B., Surya, S. and Babu, R.V., 2017. Switching convolutional neural network for crowd counting. In: *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*.

Sermanet, P., Chintala, S. and Lecun, Y., 2012. Convolutional neural networks applied to house numbers digit classification. In: *Proceedings - International Conference on Pattern Recognition*.

Shang, C., Ai, H. and Bai, B., 2016. End-to-end crowd counting via joint learning local and global count. In: *Proceedings - International Conference on Image Processing, ICIP*.

Sheng, B., Shen, C., Lin, G., Li, J., Yang, W. and Sun, C., 2018. Crowd Counting via Weighted VLAD on a Dense Attribute Feature Map. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(8).

Simonyan, K. and Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*.

Sindagi, V.A. and Patel, V.M., 2017. CNN-Based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2017*.

Sindagi, V.A. and Patel, V.M., 2018. A survey of recent advances in CNN-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107.

Sohail, A., Khan, A., Nisar, H., Tabassum, S. and Zameer, A., 2021. Mitotic nuclei analysis in breast cancer histopathology images using deep ensemble classifier. *Medical Image Analysis*, 72.

Srivastava, R.K., Greff, K. and Schmidhuber, J., 2015. Training very deep networks. In: *Advances in Neural Information Processing Systems*.

Stewart, R., Andriluka, M. and Ng, A.Y., 2016. End-to-End People Detection in Crowded Scenes. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Sun, S., Cao, Z., Zhu, H. and Zhao, J., 2020. A Survey of Optimization Methods from a Machine Learning Perspective. *IEEE Transactions on Cybernetics*, 50(8).

Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A., 2017. Inception-v4, inception-ResNet and the impact of residual connections on learning. In: *31st AAAI Conference on Artificial Intelligence, AAAI 2017*.

Van Houdt, G., Mosquera, C. and Nápoles, G., 2020. A review on the long short-term memory model. *Artificial Intelligence Review*, 53(8).

Viola, P. and Jones, M.J., 2004. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2).

Vitaly Bushaev, 2018. Understanding RMSprop — faster neural network learning. *Towards Data Science*, 36(4).

Walach, E. and Wolf, L., 2016. Learning to count with CNN boosting. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Wan, J. and Chan, A., 2019. Adaptive density map generation for crowd counting. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Wang, C., Zhang, H., Yang, L., Liu, S. and Cao, X., 2015. Deep people counting in extremely dense crowds. In: *MM 2015 - Proceedings of the 2015 ACM Multimedia Conference*.

Wang, M., Cai, H., Dai, Y. and Gong, M., 2023. Dynamic Mixture of Counter Network for Location-Agnostic Crowd Counting. In: *Proceedings - 2023 IEEE Winter Conference on Applications of Computer Vision, WACV 2023.*

Wilie, B., Cahyawijaya, S. and Adiprawita, W., 2018. CountNet: End to End Deep Learning for Crowd Counting. *Proceeding of the Electrical Engineering Computer Science and Informatics*, 5(5).

Wu, B. and Nevatia, R., 2005. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In: *Proceedings of the IEEE International Conference on Computer Vision*.

Wu, B. and Nevatia, R., 2007. Detection and tracking of multiple, partially occluded humans by Bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2).

Zeng, L., Xu, X., Cai, B., Qiu, S. and Zhang, T., 2018. Multi-scale convolutional neural networks for crowd counting. In: *Proceedings - International Conference on Image Processing, ICIP*.

Zhang, C., Li, H., Wang, X. and Yang, X., 2015. Cross-scene crowd counting via deep convolutional neural networks. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Zhang, L., Shi, M. and Chen, Q., 2018. Crowd Counting via Scale-Adaptive Convolutional Neural Network. In: *Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018*.

Zhang, X., Han, L., Shan, W., Wang, X., Chen, S., Zhu, C. and Li, B., 2023. A Multi-Scale Feature Fusion Network With Cascaded Supervision for Cross-Scene Crowd Counting. *IEEE Transactions on Instrumentation and Measurement*, 72.

Zhang, Y., Zhou, D., Chen, S., Gao, S. and Ma, Y., 2016. Single-image crowd counting via multi-column convolutional neural network. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern.*

Zhao, Z., Li, H., Zhao, R. and Wang, X., 2016. Crossing-line crowd counting with two-phase deep neural networks. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

Zou, Z., Shao, H., Qu, X., Wei, W. and Zhou, P., 2020. Enhanced 3D convolutional networks for crowd counting. In: *30th British Machine Vision Conference 2019, BMVC 2019*.

Zou, Z., Su, X., Qu, X. and Zhou, P., 2018. DA-Net: Learning the fine-grained density distribution with deformation aggregation network. *IEEE Access*, 6.