UTS UNIVERSITY OF TECHNOLOGY SYDNEY

# Convolutional Neural Network for Accurate Crowd Counting and Density Estimation

**by Saeed Amirgholipour Kasmani**

Thesis submitted in fulfilment of the requirements for the degree of

**Doctor of Philosophy**
under the supervision of Professor Xiangjian (Sean) He

University of Technology Sydney
Faculty of Engineering and IT

February 2021

# Certificate of Original Authorship Template

**Graduate research students are required to make a declaration of original authorship when they submit the thesis for examination and in the final bound copies. Please note, the Research Training Program (RTP) statement is for all students.** The Certificate of Original Authorship must be placed within the thesis, immediately after the thesis title page.

## Required wording for the certificate of original authorship

CERTIFICATE OF ORIGINAL AUTHORSHIP

I, ***Saeed Amirgholipour Kasmani*** declare that this thesis, is submitted in fulfilment of the requirements for the award of ***PhD***, in the ***SEDE/FEIT*** at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.
**If applicable, the above statement must be replaced with the collaborative doctoral degree statement (see below).*

**If applicable, the Indigenous Cultural and Intellectual Property (ICIP) statement must be added (see below).*

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 26/02/2021

## Collaborative doctoral research degree statement

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of the requirements for a degree at any other academic institution except as fully acknowledged within the text. This thesis is the result of a Collaborative Doctoral Research Degree program with ***CSIRO\Data 61***.

## Indigenous Cultural and Intellectual Property (ICIP) statement

This thesis includes Indigenous Cultural and Intellectual Property (ICIP) belonging to *[insert relevant language, tribal or nation group(s) or communities]*, custodians or traditional owners. Where I have used ICIP, I have followed the relevant protocols and consulted with appropriate Indigenous people/communities about its inclusion in my thesis. ICIP rights are Indigenous heritage and will always remain with these groups. To use, adapt or reference the ICIP contained in this work, you will need to consult with the relevant Indigenous groups and follow cultural protocols.

# ABSTRACT

**Convolutional Neural Network for Accurate Crowd Counting and Density Estimation**

by

Saeed Amirgholipour Kasmani

Nowadays, crowd and object counting has become an important task for a variety of applications, such as traffic control, public safety, urban planning, and video surveillance. It has also become a crucial part of building a high-level monitoring system such as video surveillance and crowd analysis. In these cases, dynamic crowd monitoring and analysis is extremely important for control management and social safety.

Like the other computer vision issues, crowd counting and density estimation come with various kinds of challenges such as high clutters, occlusions, non-uniform distributions of objects or people, and intra-scene and inter-scene variations in appearance. Researchers and industrial partners have attempted to design and develop many sophisticated models to address various issues that exist in crowd counting. Especially in recent years, the number of researches in the crowd counting era became overwhelming with the domination of deep-learning and Convolution Neural Networks (CNNs) based models in various computer vision tasks. In this thesis, we revisit the crowd counting and propose various novel solutions to this problem.

At first, we propose an Adaptive Counting Convolutional Neural Network (A-CCNN) and consider the scale variation of objects in a frame adaptively to improve the accuracy of counting. Our method takes advantages of contextual information to provide more accurate and adaptive density maps and crowd counting in a scene. Then, we focus on CNN pruning to further enhance the crowd counting models for real-time application and increase the performance of CCNN model. Thus, a new

pruning strategy is proposed by considering the contributions of various filters to the final result. The filters in the original CCNN model are grouped into positive, negative, and irrelevant types. We prune the irrelevant filters, of which feature maps contain little information, and the negative filters determined by a mask learned from the training dataset. Our solution improves the results of the counting model without fine-tuning or retraining the pruned model. Finally, we propose a novel Pyramid Density-Aware Attention-based network, abbreviated as PDANet, which leverages the attention, pyramid scale feature and two branch decoder modules for density-aware crowd counting. The PDANet utilises these modules to extract different scale features, focus on the relevant information, and suppress the misleading ones. Extensive evaluations conducted on the challenging benchmark datasets well demonstrate the superior performance of the proposed models in terms of the accuracy of counting as well as generated density maps over the well-known state-of-the-art approaches.

Dissertation directed by Professor Xiangjian (Sean) He
School of Electrical and Data Engineering

# Dedication

I owe thanks to a very special person, my wife, Mozhgan for her continued and unfailing love, support and understanding during my pursuit of Ph.D degree that made the completion of thesis possible. This thesis work is dedicated to you who were always around at times I thought that it was impossible to continue, and you helped me to keep things in perspective. I greatly value her contribution and deeply appreciate her belief in me.

This work is also dedicated to my mother, Soraya, who had always loved me unconditionally and whose good examples have taught me to work hard for the things that I aspire to achieve.

# Acknowledgements

At this moment of accomplishment, I would like to express my deep appreciation to my wonderful supervisor, Prof Sean He, who has been guiding, supporting, and encouraging me throughout my PhD candidature journey. He gave me care and love like a father, which I can never forget. I am also greatly indebted to my co-supervisor, Dr Wenjing Jia, for her involvement and guidance throughout my PhD. She offered me her positive appreciations, constructive criticisms, and valuable advices during my PhD research which helped me tackle the challenges and complete the research work. I extend thanks to Prof Dadong Wang, my co-supervisor from CSIRO/DATA 61, for his time, encouragement, and expertise throughout this journey. I was able to accomplish this research due to the tremendous and unforgettable supports from my supervisory panel.

I need to express my deep gratitude to UTS and CSIRO/ Data61 to sponsor support and scholarships. I am thankful to the University of Technology Sydney, to give me permission to conduct PhD studies and provide me with state-of-the-art facilities to do my experiments. I am thankful to all staff at the School of Electrical and Data Engineering, FEIT, and GRS staffs of UTS for their kind assistance and support. It is my fortune to acknowledge my friends' support throughout the research tenure gratefully.

In the end, my wife, my family, and my parents, I sincerely thank you for believing me and showing faith in me. I would not have been able to focus on my research without the ongoing support and unconditional love of my wife.

Saeed Amirgholipour Kasmani

Sydney, Australia, 2021.

# List of Publications

### Journal Papers

J-1. **Amirgholipour**, S., Jia, W., Liu, L., Fan, X., Wang, D. and He, X., 2021. PDANet: Pyramid Density-aware Attention Net for Accurate Crowd Counting. Neurocomputing, 451, pp.215-230.

J-2. Liu, L., Jia, W., Jiang, J., **Amirgholipour**, S., Wang, Y., Zeibots, M. and He, X., 2020. Denet: A universal network for counting crowd with varying densities and scales. IEEE Transactions on Multimedia, 23, pp.1060-1068.

J-3. Liu, L., **Amirgholipour**, S., Jiang, J., Jia, W., Zeibots, M. and He, X., 2019. Performance-enhancing network pruning for crowd counting. Neurocomputing, 360, pp.246-253.

### Conference Papers

C-1. **Amirgholipour**, S., He, X., Jia, W., Wang, D. and Zeibots, M., 2018, October. A-CCNN: Adaptive CCNN for density estimation and crowd counting. In 2018 25th IEEE International Conference on Image Processing (ICIP) (pp. 948-952). IEEE.

# Contents

# List of Figures

# Abbreviation

CNN - Convolutional Neural Network

MAE - Mean Absolute Error

RMSE - Root Means Square error

Weak-Sup: Weak Supervised

Self-Sup: Self Supervised

Semi-Sup: Semi Supervised

Un-Sup: Un Supervised.

Fully-Sup: Fully Supervised

FIb: Full Image-Based

Pb: Patch-based

DOF: Degree Of Freedom

HPs: Hyper Parameters

# Nomenclature and Notation

Capital letters denote matrices.

Lower-case alphabets denote column vectors.

$(.)^T$ denotes the transpose operation.

$I_n$ is the identity matrix of dimension $n \times n$.

$0_n$ is the zero matrix of dimension $n \times n$.

$\mathbb{R}$, $\mathbb{R}^+$ denote the field of real numbers, and the set of positive reals, respectively.

# Chapter 1

# Introduction

Nowadays, crowd and object counting has become an important task for a variety of applications, such as traffic control [88], public safety, urban planning, and video surveillance [127, 187]. Many studies have been done on various domains such as city traffic control and vehicle counting [107, 177, 181, 41], people counting [183, 107, 8, 63, 123, 136, 81, 49, 179, 124, 9, 76, 150], farm and animal counting [3], environment resources for sustainability [32, 173], bio and cell counting [158, 152, 71] and leave counting in plant phenotype [1, 39]. It has also become a crucial part of building a high-level monitoring system such as video surveillance [11] and crowd analysis [127, 187]. It is due to the overwhelming growth of world population and urbanization, with many scenarios where crowd gatherings to participate in an event e.g. stadiums, concerts, and parades. Thus, dynamic crowd monitoring and analysis is extremely important for crowd management and public safety.

## 1.1 Background

Due to the indispensable role of crowd counting in various scenarios, it has attracted more and more interests from research and industrial community. They have attempted to design and develop many sophisticated models to address various issues that exist in crowd counting. Especially in recent years, the number of research in the crowd counting area has become overwhelming by domination of Convolution Neural Networks (CNNs) based architectures in numerous computer vision tasks. Despite the fact that these tasks are distinctive, some common attributes exist, such as distribution patterns and structural features. Crowd counting techniques

are mainly applied for people and car counting. However, they can also be used in other fields. The focus of this thesis is on CNN-based crowd counting solutions.

### 1.1.1 Related Works and Scope

Generally, crowd counting approaches are classified into two major groups: detection based approaches and regression or density based methods. However, in recent years, thanks to the advancement of deep learning technologies, CNN-based regressors and density estimators have become the main stream. This chapter focuses on CNN-based approaches due to the excellent performance of the CNN-based crowd density estimation. However, to complete the picture, it is essential to know about some other related studies in this subsection.

Detection-based approaches [147, 74, 70, 28] were the early trend in the crowd counting area. These methods utilise the patch-based approaches to detect a person or head in an image. Although the recent superb object detectors such as YOLO-v5 [112], RCNN [38, 115, 46], and SSD [87], can achieve outstanding detection performance in the low crowdedness and sparse scenes, they often produce disappointing results in more complex situations with background clutter and occlusion in immensely dense crowds.

To address the occlusion and overcrowded issues, many researchers [11, 55, 12] have focused their approaches on the regression-based approaches, which count from the image patches directly. These approaches first generate local features [119] (e.g., LBP [103], HOG [23], SIFT [92], GLCM [44]) or global features [14] (such as texture, gradient, edge features). They then utilise the regression-based machine learning techniques such as Gaussian mixture regression [145] or linear regression [108] to learn to estimate and count the number of people.

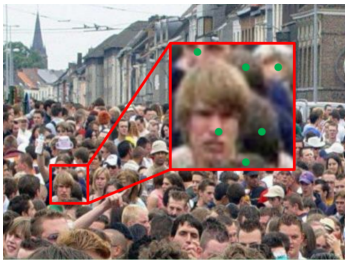
These methods have been found successful when dealing with problems like occlusion and background clutter. However, spatial information is what they always

ignore. As a result, Lemptisky et al. [71] first proposed a model based on density estimation to learn density maps linearly. To avoid the challenge of learning a linear mapping, [109] presented a regression-based random forest, which has achieved a decent performance by utilising it to train various forests and introducing a initial crowdedness level detection. Moreover, less memory is required in this method to store the forest. The spatial information is considered in these methods, yet only old school features are used to obtain low-level crowdedness features. But, it fails to produce the high-quality feature map for a robust crowd counting.

With the emerging CNNs concept as a powerful feature detector in the image processing domains, more researchers used it to boost the estimation performance in the crowd counting application. Premature approaches took the benefit of a basic and straightforward CNNs to estimate the density of the crowds [33, 154, 176, 152], which achieved remarkable improvement compared with the previous hand-crafted local or global features. However, soon after that many researchers realized that utilizing the Fully Convolution Network (FCN) could be a more efficient and effective solution, and it had become the new mainstream and trends for crowd counting and density estimation. Due to the aim of this thesis, we only focus on the CNN-based models that deliver crowd counting and density estimation solutions.

## 1.2 Challenges

Crowd counting is a tough and interesting research area, due to lots of existing challenges, such as occlusion, non-uniform distribution, perspective distortion, illumination variation, complex background, rotation, scale variation and changing in weather condition [35]. Some sample images are illustrated in Fig. 1.1. Besides, there are the other challenges, such as indoor, outdoor, or in the wild scenes. These challenges are not mutually exclusive, and this fact makes the crowd counting become a more complex task.



(a) Occlusion

(b) Non-uniform distribution

(c) Perspective distortion

(d) Illumination variation

(e) Complex background

(f) Rotation

(g) Weather changes

(h) Scale variation

Figure 1.1 : **The existing tough challenges in the crowd counting task [35].**

• **Occlusion**. When the density of the crowd increases, people appear to occlude each other partly. Occlusion directly affects the crowd counting solutions because it is difficult for the models to detect each individual and generate the accurate density maps.

• **Non-uniform distribution.** As shown in the Fig. 1.1, the density of crowd can vary from scene to scene and within the local regions in the same scene. Thus, Having a diverse global and local density and a large distribution variation is common for the benchmark crowd counting datasets.

• **Perspective distortion.** Fig. 1.1 privodes an extreme scale variation senario which can emerge with the perspective distortion. In many occasions in the crowd counting, there is not any information about a camera setup for verifing the six degree-of-freedom (DOF) of the camera, which makes the perspective distortion be a real challenge for crowd counting tasks.

• **Illumination variation.** The illumination can change among various scenes, or even in the same scene. Even worse, it also changes at different times in a day.

• **Complex background.** It is also common to have background regions with similar patterns and colours with the foreground.

• **Rotation.** As illustrated in Fig. 1.1, an extreme rotation variation can occur due to the camera position and photographic angles.

• **Weather changes.** In many cases, there are some fixed cameras that capture scenes in the different day under different weather conditions, e.g. sunny, clear, foggy, and rain.

• **Scale variation.** The scale variation is one of the most critical issues that almost all crowd counting solutions have to face, specifically for those models based on density estimation.

As mentioned above, the challenges prove why crowd counting has become a challenging task and still has rooms for much research. It also indicates a path for researchers to propose a solution to address these issues.

## 1.3 Research Specification

The current research targets specific Stakeholders and objectives. The following subsections provide lists of potential stakeholders and objectives, significance, and ethics and risk consideration for this research.

### 1.3.1 Research Stakeholders and Objectives

***Stakeholders***

The stakeholders of our research include:

1. Engineers from technology companies: those companies pay great attention to solve real-world problems through developing artificial intelligence algorithms and models.

2. Research communities or researchers whose topic is related to social scene understanding, deep learning or computer vision.

3. Information based manufacturers: those manufacturers are aiming to design smarter social media systems.

***Research Objectives***

The objectives of this research are:

1. Understanding: To understand how CNNs extract features from input images and what the relationship of these features is with our target applications about crowd counting and density estimations.

2. Improving: To improve the performance of crowd counting methods with an adaptive CNNs based crowd counting model, and novel network pruning methods.

3. Innovation: To propose novel models for automatically understanding the crowded scene based on innovative deep learning methods.

### 1.3.2 Research Significance

At first, social scene understanding is required in many real-world scenarios. For example, if we can obtain high-level information about a scene such as finding different groups, activities and suspicious actions, it can be much easier to manage many surveillance situations. Nowadays, more and more technology companies pay attention to artificial intelligence fields. They are committed to applying research outcomes to real-world applications and developing products much 'smarter' than ever before. Therefore, we believe the proposed end-to-end scene image understanding system has an excellent prospect for real-world applications. The engineers from technology companies should be interested in these work.

Last but not the least, the security and surveillance industry has made great progress in past decades. People are no longer satisfied with the traditional systems for simple detection of objects. They hope to develop more intelligent systems to understand high-level information same as the human. However, it is a challenge to transfer the existing AI algorithms which perform well on computers to image understanding systems (i.e., group activity analysis systems). We hope our works are meaningful to big AI and social media based manufacturers who aim to develop smarter systems. Thus, some of the applications of crowd counting and density estimation are given bellow.

1. Crowd control and management: Nowadays, there are enormous number of surveillance cameras around the world. Crowd counting and density estimation

can help us deliver better crowd management and assure the public safety better than the past during emergencies such as riot, fire, and stampede.

2. Transportation system design and traffic control: With the help of people and car counting, we are able to have a better picture about the traffic distributions in the city, and therefore to address these critical situations in real time.

3. Public Space Design: Crowd counting and analysis provides valuable information to architects and civil engineers, and assist them with the design of public spaces.

4. Counting cells or bacteria on the microscopic level. With viable cell count, researchers and scientists can detect the number of bacteria divisions growing in a sample. Then, they can use this information to identify why a patient has a particular issue, and how they can react to the problems, and help the patient.

5. Visual Surveillance: Recently, the application of crowd counting for public safety has become clearer due to the unprecedented COVID-19 pandemic. With the help of crowd counting and crowd analysis we are able to track crowd and monitor social distancing in various events and situations.

6. Natural resource monitoring: Density estimation based methods help us to process the satellite images and monitor our natural resources such as forest and protect them unexpected disaster.

7. Intelligent Environment: The intelligence can be used in a shopping mall to find the pattern of customers and their interest in a particular object.

### 1.3.3 Ethics and Risk Consideration

There is no foreseeable risk of any harm for this research.

## 1.4 Contributions

In this thesis, we investigate the applications of deep learning technologies in the crowd counting and density estimation. At first, we will try to thoroughly understand how CNNs address the crowd counting and density estimation and then we propose a new solution to learn strong features, and decrease the effect of negative and irrelevant features. In the next stage, we aim to improve the performance of the state-of-the-art approaches with adaptive selecting and changing hyper-parameters for training convolutional models. Finally, we utilise our experiences to propose novel solutions for CNN-based crowd counting and density estimation. These solutions address the challenges in the crowd counting in a novel ways. In the following sections, we will summarise the most significant innovation and contributions of this research.

### 1.4.1 A-CCNN: Adaptive CCNN for Density Estimation and Crowd Counting [2]

Crowd counting, used for estimating the number of people in a crowd using vision-based computer techniques, has attracted much interest in the research community. Although many attempts have been reported, real-world problems, such as huge variation in subjects' sizes in images and serious occlusion among people, make it still a challenging problem. We propose an Adaptive Counting Convolutional Neural Network (A-CCNN) and consider the scale variation of objects in a frame adaptively so as to improve the accuracy of counting. Our method takes advantages of contextual information to provide more accurate and adaptive density maps and crowd counting in a scene. Extensively experimental evaluation is conducted using different benchmark datasets for object-counting and shows that the proposed approach is effective and outperforms state-of-the-art approaches.

### 1.4.2 Performance-Enhancing Network Pruning for Crowd Counting [82]

The Counting Convolutional Neural Network (CCNN) has been widely used for crowd counting. However, they typically end up with a complicated network model resulting in a challenge for real-time processing. The existing solutions aim to reduce the size of the network models, but unavoidably sacrifice the accuracy. Different from the existing pruning solutions, a new pruning strategy is proposed by considering the contributions of various filters to the final result. The filters in the original CCNN model are grouped into positive, negative and irrelevant types. We prune the irrelevant filters of which feature maps contain little information, and the negative filters determined by a mask learned from the training dataset. Our solution improves the results of the counting model without fine-tuning or retraining the pruned model. We demonstrate the advantages of our proposed approach on the problem of crowd counting. Our experimental results on benchmark datasets show that the network model pruned using our approach not only reduces the network size but also improves the counting accuracy by 4% to 17% less MAE than the state of the arts.

### 1.4.3 PDANet: Pyramid Density-aware Attention Net for Accurate Crowd Counting

Although many attempts have been reported, crowd counting remains an open real-world problem due to the vast scale variations in crowd density within the interested area, and severe occlusion among the crowd. In this solutions, we propose a novel Pyramid Density-Aware Attention-based network, abbreviated as PDANet, that leverages the attention, pyramid scale feature and two branch decoder modules for density-aware crowd counting. The PDANet utilises these modules to extract different scale features, focus on the relevant information, and suppress the misleading ones. We also address the variation of crowdedness levels among different

images with an exclusive Density-Aware Decoder (DAD). For this purpose, a classifier evaluates the density level of the input features and then passes them to the corresponding high and low crowded DAD modules. Finally, we generate an overall density map by considering the summation of low and high crowded density maps as spatial attention. Meanwhile, we employ two losses to create a precise density map for the input scene. Extensive evaluations conducted on the challenging benchmark datasets well demonstrate the superior performance of the proposed PDANet in terms of the accuracy of counting and generated density maps over the well-known state of the arts.

## 1.5 Thesis Organization

This thesis is organised as follows:

- *Chapter 1* presents an introduction to the research topic, aims, and related studies and scope of the research, its significance, major challenges, its objectives, and outline the contributions to the research.

- *Chapter 2* provides a literature review around the crowd counting topics. Due to the domination of CNN-based models in recent years, we mostly review the modern CNN-based solutions. We investigate the CNN-based crowd counting and density estimation approaches by various criteria, such as their network architectures, learning paradigm, inference manner, and their supervision forms. We also provide a comprehensive list of crowd counting datasets and group them in different categories based on their usages and applications. Then, we present two main performance-evaluation metrics for crowd counting. Finally, we compare the state-of-the-art crowd counting solutions according to the six most cited benchmark datasets. We also extract the main properties of the dominant approaches in each dataset and explain why some features are

unique and can significantly affect the performance of crowd counting models.

- *Chapter 3* presents our first solution in the crowd counting research area. This solution is based on one of the best crowd counting method in 2016 [107], CCNN. We have found some gaps in [107], so we try to address them with A-CCNN: Adaptive CCNN for density estimation and crowd counting.

- *Chapter 4* investigates the effect of neural network prunning on crowd counting. In this chapter, we evaluate the impact of each channels on the density maps generation, and then devide them to various categories. Finally, we prune them based on their significant on the crowd counting results.

- *Chapter 5* provides our last end-to-end solutions for density estimation and crowd counting. In this solutions, we propose a novel Pyramid Density-Aware Attention-based network, abbreviated as PDANet, which leverages the attention, pyramid scale feature and two branch decoder modules for density-aware crowd counting.

- *Chapter 6* presents a brief summary of the thesis contents and its contributions are given in the final chapter. Recommendation for future works is given as well.

# Chapter 2

# Literature Review

As discussed in Chapter 1, crowd counting is becoming so popular in recent years for various applications. However, intra-scene and inter-scene scale, perspective variations, and luminous fluctuation within the image, low resolution data, non-uniform density, and severe occlusions have made the counting and density map estimation be a tough and challenging task [35]. Thus, many researchers are attracted to work on this task to address these challenges, which has yielded various high quality literature and novel solutions. Despite of focusing on counting the number of objects within a low crowdedness scene in the early stage, the state-of-the-art models mostly focus on handling high density variations in real world crowd scenes. In this study, we aim to understand where and why some state-of-the-art methods work more effectively in some situations, and what characteristics help these models to reach their high performance. The aim of this chapter is to characterize the key components that have most impact on crowd density estimation and propose some suggestions regarding the future study based on these findings. Therefore, a survey on the state-of-the-art solutions is conducted along with the key methods in the crowd density estimation research area. Due to the huge popularity of the CNN based techniques emerged since 2013, we just focus on the solutions based on the CNN techniques. To prepare this survey, we refer to the survey structure, figures and tables of Gao et al. [35].

## 2.1 Introduction

In this chapter, at first, we list the most important crowd counting solutions. Then, we define some discriminating characteristics for crowd density estimation solutions. After providing a list of most important crowd counting datasets, we evaluate the techniques on benchmark datasets. Finally, based on the experimental results and their merits and drawbacks, we propose the top key effective aspects for a well-established future crowd counting and estimation model.

Defining the scope of the literature review, in this section, our attention is paid onto those well-established algorithms, which are all indispensable or inspiring papers published in premium conferences and journals. We focus on the recent neural network based models for crowd density estimation. Moreover, to cover the every aspect, we also review those most important earlier works. We categorise the existing approaches by considering their supervision form, network architecture, etc. These systematic taxonomies and comprehensive review help us to draw a clearer picture and acquire better understanding about the innovation of crowd density estimation models in the studies since 2013.

### 2.1.1 Organisation of This Chapter

To summarize, the tasks of this chapter following and referring to the survey of Gao et al. [35] are as follows.

- We show a systematic and extensive review about the modern CNN-based crowd counting models, which are classified according to several criteria, including the network architectures used, supervised or unsupervised approaches, learning criteria, etc.

- We provide a comprehensive survey on the existing datasets in the areas of crowd and instance counting. These datasets are grouped into several cate-

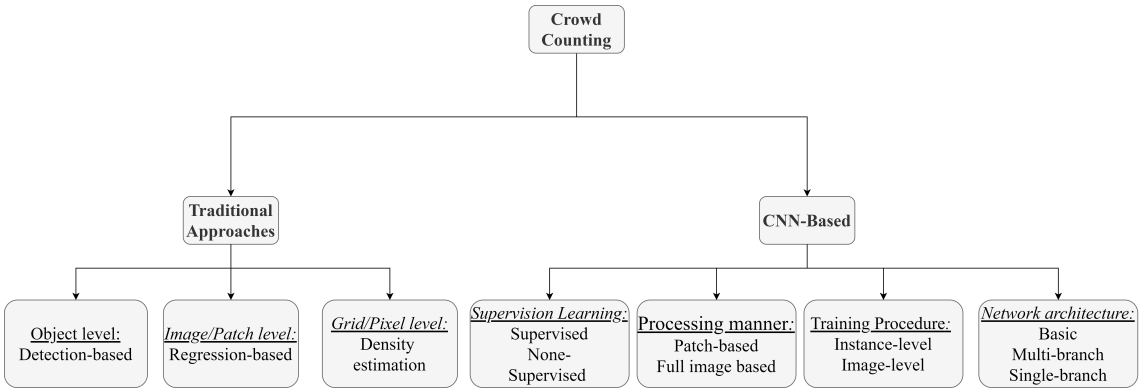



Figure 2.1 : **High level Classification of Crowd Counting solutions [35].**

gories, i.e., the most frequently used datasets, the recently introduced datasets, the special crowd counting datasets, and the object counting datasets.

- We introduce two mostly used image-level evaluation metrics, i.e., Root Mean Squared Error (MSE) and Mean Absolute Error (MAE) [9, 76, 15, 88].

- We demonstrate the performance of the state-of-the-art methods based on the most-widely used benchmark datasets. These comprehensive experiments help us draw a picture of the impact of each approach on the crowd counting and density estimation results.

The rest of the chapter is organised as follows. In Section 2.2, we provide a survey of the major CNN-based crowd estimation solutions according to several different taxonomies. Then, we investigate the existing datasets in crowd and the object counting areas in Section 2.3. In Section 2.4, we present metrics that can be used for the performance evaluation of crowd counting models. Finally, we investigate the performance of the existing methods based on the most frequently used benchmark datasets in Section 2.5.

## 2.2 Key Characteristic of Crowd Counting Solutions

In this part of the literature review, we investigate the CNN-based crowd counting and density estimation approaches based on various criteria as shown in Fig. 2.1 (Subsection 2.2.4). Mainly, we review the network architectures of the crowd counting models. Then, in Subsection 2.2.2, we survey the learning paradigm of the models (Subsection 2.2.3). Afterwards, we divide the methods by their network inference manners and how they process the input data (Subsection 2.2.1). Besides, we review the networks by their supervision forms.

In the following subsection, we consider the well-established models in the groups and briefly explain them. We recall the information from the survey paper [35] of Gao et al.with our own critical comments. Table 2.1 provides a Characteristic summary of the state-of-the-art crowd density estimation solutions.

Table 2.1 : **Characteristic summary of state-of-the-art CNN-based crowd counting approaches [35].**

| Methods | Supervision learning | Training Procedure | Processing manner | Network architecture | Venue | Year |
|---|---|---|---|---|---|---|
| Fu et al. [33] | Fully-Sup. | STT | Patch-based | Basic | EAAI | 2015 |
| Cross scene [176] | Fully-Sup. | MTT | Patch-based | Basic | CVPR | 2015 |
| Wang et al. [154] | Fully-Sup. | STT | Patch-based | Basic | ACMMM | 2015 |
| MCNN [183] | Fully-Sup. | STT | full image-based | Multi-branch | CVPR | 2016 |
| CNN-Boosting [152] | Fully-Sup. | STT | Patch-based | Basic | ECCV | 2016 |
| Crowdnet [8] | Fully-Sup. | STT | Patch-based | Multi-branch | ACMMM | 2016 |
| Shang et al. [126] | Fully-Sup. | STT | full image-based | Multi-branch | ECCV | 2016 |
| Hydra-CNN [107] | Fully-Sup. | MTT | Patch-based | Multi-branch | ECCV | 2016 |
| Switching CNN [123] | Fully-Sup. | MTT | Patch-based | Multi-branch | CVPR | 2017 |
| CMTL [135] | Fully-Sup. | MTT | full image-based | Multi-branch | AVSS | 2017 |
| CP-CNN [136] | Fully-Sup. | MTT | full image-based | Multi-branch | ICCV | 2017 |

**Table 2.1 continued from previous page**

| Methods | Supervision learning | Training Procedure | Processing manner | Network architecture | Venue | Year |
|---|---|---|---|---|---|---|
| CSRNet [76] | Fully-Sup. | STT | full image-based | Single-branch | CVPR | 2018 |
| SaCNN [179] | Fully-Sup. | MTT | full image-based | Single column | WACV | 2018 |
| DecideNet [81] | Fully-Sup. | MTT | Patch-based | Multi-branch | CVPR | 2018 |
| D-ConvNet [178] | Fully-Sup. | STT | full image-based | Single-branch | CVPR | 2018 |
| DRSAN [85] | Fully-Sup. | STT | full image-based | Multi-branch | IJCAI | 2018 |
| SACNN [9] | Fully-Sup. | MTT | Patch-based | Single column | ECCV | 2018 |
| ACSCP [128] | Fully-Sup. | MTT | Patch-based | Multi-branch | CVPR | 2018 |
| IG-CNN [5] | Fully-Sup. | MTT | Patch-based | Multi-branch | CVPR | 2018 |
| CL [56] | Fully-Sup. | MTT | Patch-based | Single-branch | ECCV | 2018 |
| NetVLAD [133] | Fully-Sup. | MTT | full image-based | Single-branch | TII | 2018 |
| GAN-MTR [105] | Semi-Sup. | MTT | full image-based | Basic | WACV | 2018 |
| ic-CNN [111] | Fully-Sup. | MTT | full image-based | Multi-branch | ECCV | 2018 |
| L2R [89] | Self-Sup. | MTT | full image-based | Basic | CVPR | 2018 |
| SPN [15] | Fully-Sup. | STT | full image-based | Single column | WACV | 2019 |
| PaDNet [146] | Fully-Sup. | STT | Patch-based | Single-branch | TIP | 2019 |
| SR-GAN [106] | Semi-Sup. | MTT | full image-based | Basic | CVIU | 2019 |
| ASD [162] | Fully-Sup. | MTT | full image-based | Multi-branch | ICASSP | 2019 |
| SAAN [49] | Fully-Sup. | MTT | full image-based | Multi-branch | WACV | 2019 |
| SE Cycle GAN [156] | Fully-Sup. | STT | full image-based | Single column | CVPR | 2019 |
| SAA-Net [150] | Fully-Sup. | MTT | full image-based | Single column | CVPR | 2019 |
| SFCN†[2] [156] | Fully-Sup. | STT | full image-based | Single column | CVPR | 2019 |
| ADCrowdnet [86] | Fully-Sup. | STT | full image-based | Single column | CVPR | 2019 |
| CAN&ECAN [88] | Fully-Sup. | STT | full image-based | Single column | CVPR | 2019 |
| PCC Net [36] | Fully-Sup. | MTT | full image-based | Multi-branch | TCSVT | 2019 |
| PACNN [130] | Fully-Sup. | STT | full image-based | Single column | CVPR | 2019 |
| W-Net [149] | Fully-Sup. | STT | full image-based | Single column | CVPR | 2019 |
| CFF [131] | Fully-Sup. | MTT | full image-based | Single-branch | ICCV | 2019 |
| SL2R [90] | Self-Sup. | MTT | full image-based | Basic | CVPR | 2019 |

**Table 2.1 continued from previous page**

| Methods | Supervision learning | Training Procedure | Processing manner | Network architecture | Venue | Year |
|---|---|---|---|---|---|---|
| RReg [153] | Fully-Sup. | STT | full image-based | Multi-branch | CVPR | 2019 |
| SFANet [188] | Fully-Sup. | MTT | full image-based | Single column | CVPR | 2019 |
| RAZNet [80] | Fully-Sup. | MTT | full image-based | Multi-branch | CVPR | 2019 |
| TEDnet [61] | Fully-Sup. | STT | full image-based | Single column | CVPR | 2019 |
| HA-CCN [138] | Weak-Sup. | STT | full image-based | Single column | TIP | 2019 |
| AT-CNN [185] | Fully-Sup. | MTT | full image-based | Single-branch | CVPR | 2019 |
| L2SM [166] | Fully-Sup. | STT | Patch-based | Single column | ICCV | 2019 |
| McML [18] | Fully-Sup. | STT | full image-based | Multi-branch | ACM MM | 2019 |
| GWTA-CCNN [122] | Un-Sup. | STT | Patch-based | Single column | AAAI | 2019 |
| ILC [19] | Fully-Sup. | MTT | full image-based | Multi-branch | CVPR | 2019 |
| RANet [174] | Fully-Sup. | STT | full image-based | Multi-branch | ICCV | 2019 |

### 2.2.1 Model Supervision Method

We can divide the crowd counting solutions to two main categories due to their need for annotated data, i.e., none-supervised learning and fully-supervised learning models.

Most CNN-based crowd counting models utilise the annotation data to generate density maps and count objects. However, due to the difficulty of providing the annotation data, the crowd counting models have the overfitting problem, which leads to a noticeable decrease in their capabilities for more practical applications. Thus, taking advantage of the weakly labelled or unlabeled data seems to be a promising research area shortly.

None-supervised, unsupervised and semi-supervised learning methods learn without or with a few labelled data. However, self-supervised models add extra tasks to help the models learn annotation by themselves. Recently, some methods take

advantage of the unlabelled images in training time and have demonstrated excellent results relatively with fully supervised models. Almost all of the unsupervised or weak-supervised approaches are based on GAN, in fact researchers combine GAN with well-established CNN to provide unsupervised or semi supervised crowd counting solutions. Olmschenk et al.proposed the first GAN based models for crowd counting [105]. GAN-MTR [105] utilises semi-supervised learning GANs for various object regression problems, and it uses the unlabeled data for training of a basic model [176]. They furthur improved the idea by proposeing SR-GAN [106] and DG-GAN [104]. SR-GAN [106] introduces semi-supervised model based on GANs with a novel feature contrasting loss function for crowd counting. DG-GAN [104] is another framework that uses a semi-supervised GAN based architecture to find each individual in the crowd scene. GWTA-CCNN [122] is an semi-supervised crowd counting which combine CCNN [107] with GAN. It deploys an autoencoder, and relies on Grid Winner-Take-All [98] method to learn crowd characteristics and features unsupervised manner.

HA-CNN [138] and L2R [89, 90] are two semi/self supervised models which do not use GAN in their models. HA-CNN [138] fine-tunes the network with a weakly supervised learning approach, by extracting the density level information from the image-level labels. L2R [89, 90] extracts pyramid patches from the input image, then knowing that the number of people in the inside patch is less or equal to the crowd in the larger patch, it learns a pre-training CNN in a semi-supervised manner. However, L2R is a fundamentally supervised method, but it defines an additional count ranking task based on a self-supervised learning approaches.

### 2.2.2 Model Training and Learning Procedure

There are two primary learning methods for crowd counting solutions, i.e., the single-task and multi-task approaches in view of different types of learning ap-

proaches.

The most straight forward methods for learning are the single-task learning [10]. Most CNN-based crowd counting methods learn to generate density maps, and then crowd count number comes from the summation of all pixels, or the initial works learn to count the exact number from the image directly. Albeit the single-task learning (specially density-based models) can achieve good performance, they still have some issues with the scale and density variation within and among various input scenes. Most recently, multi-task learning models have been proposed to address these issues by considering some extra information. Multi-task learning tends to come with multiple subset models. Each subset can be Single-branch or multiple branch architectures. Many studies have been done in this area [135, 81, 135, 5, 36]. Sindagi et al. proposed CMTL [135], which utilises a combination of regression and classification to propose a new multi-task learning solution for crowd density estimation. It classifies the input feature maps and uses this additional information to estimate final density maps. Decidenet [81] is an intresting research that combine the detection and regression. Decidenet proposes a conditional network structure that adaptively switches between two alternative counting models, counting by the detection or by regression. It takes advantage of a classification module to help the network set appropriate hyperparameters and relative weights for selecting the proper counting mode. The Multi-branch structure of this model is the drawback of this model. Ranjan et al.proposed ic-CNN [111] which is a two-branch architecture that generates the low-resolution density maps in one branch and then passes this information plus the feature maps generated from the previous layers to the second branch to refine the density map and generate higher quality map for the input image. RAZ-Net [80] is proposed by Liu et al.in 2019. It addresses the inconsistency issue between crowd localization and the density map. It defines a zooming network to improve the resolution, and localize the crowd effectively. In addition, it defines

an adaptive fusion model for augmenting the compatibility between localization and crowd counting.

### 2.2.3 Processing Manner

There are two major methods to process the input scene for the training of the CNN-based crowd counting solutions, i.e., patch based and full image based inference.

The full image based Processing approaches only consider the entire input scene as the input and generate the final density map based on that. However, they tend to miss some local information. Most of the crowd counting models are based on full image based processing [129, 126].

On the other hand, the patch based Processing methods need to extract random patches from the training data. In the validation step, patch-based methods utilise sliding windows to extract patches and feed them to the model and finally assembles the model results to produce the final density map and count. Patch-based solutions have two major problems, i.e., the computational cost due to sliding window operation and the ignorance of the global information.

Cross-scene was among the first patch based processing model which was introduced by Zhang et al.in 2015 [176]. Cross-scene extracts intersecting patches and feed them as training samples. On the other hand, it considers the density map of corresponding patch as the annotated label. One of the most successful patched based models was CCNN [107]. CCNN is based on extracting fix-size patches and feeding them to the models. CCNN uses a Gaussian function for generating the ground truth density map, and it also utilises various covariance values for the Gaussian function for the various datasets.

Inspired by the Hydra CNN [107] method, some researchers have tried to utilise

different deep models to solve the problem caused by the significant variance of crowd's appearance in a captured image/video. Deepak et al. [123] proposed a switching CNN to select the best CNN regressor for each of different receptive fields and achieved better results than the state-of-the-arts for crowd counting. Kumagai1 et al. [67] proposed a mixture of Counting CNNs and adaptively selected multiple CNNs according to the appearance of a test image for predicting the number of people. Zhang et al. proposed a multi-column network and three independent CNN architectures, and then used the combined features of these three networks to get a density map [172].

In recent years a number of studies have been done on patch based processing [157, 146, 166, 165]. DML [157] combined metric learning with a CNN and extract density-level features from each patch and uses this information for measuring the distance in density map creation time. PaDNet [146] proposes a Density-Aware Network (DAN) for addressing the density variation of the crowd. Besides, it presents a Feature Enhancement Layer (FEL) module to improve local and global density estimation performance. and finally L2SM [166, 165] introduces Scale Preserving Network (SPN) and a learn-to-scale-module (L2SM) to handle the intra density changes in the input images. These two modules help to get patch-level density maps and to compute scale ratios for dense regions in each patch, correspondingly.

### 2.2.4 Crowd Counting Network Architectures

Finally, we present the most important characteristic of crowd counting solutions. We can classify the CNN-based architectures into three major categories, i.e., the basic Architecture, multi-branch Architecture, and single-branch Architecture. Fig. 2.2 illustrates the architectures of the three main categories.

**Basic Architecture** utilising the primary pre-trained networks or CNNs with their convolution, pooling, and fully connected layers to count objects in the scene.

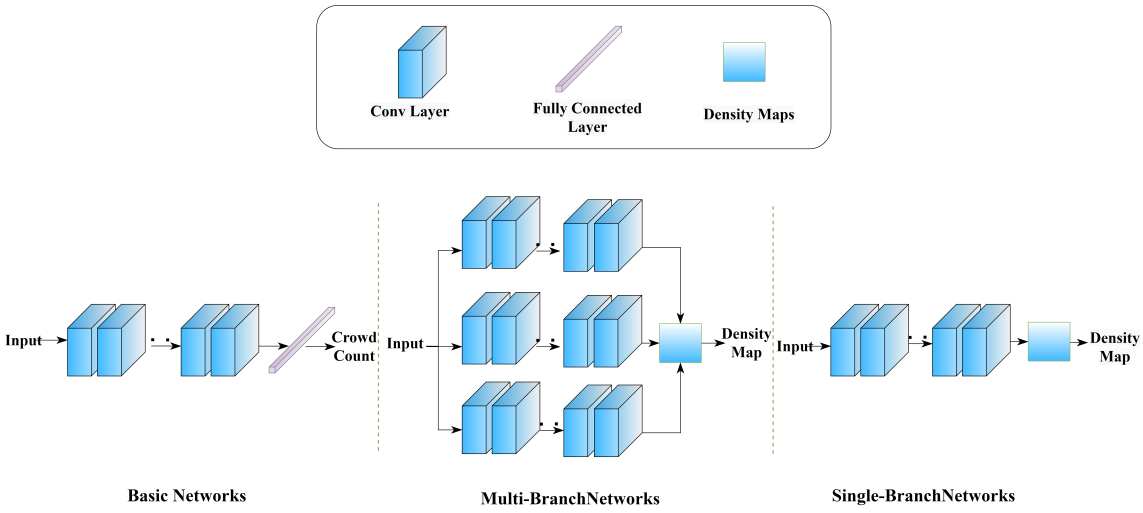



Figure 2.2 : **Three different architectures for existing crowd counting approaches [35].**

Initial works based on CNNs employed them for crowd density estimation and counting. The basic CNN models are easy to deploy, but their crowd counting accuracy are lower than those of the more complicated CNN based models. Fu et al. [33] introduce the first crowd counting method based on this technique, which improves the speed and accuracy of crowd counting by eliminating some connections and putting two layers of ConvNet classifiers at the end of the model. Wang et al. [154] were the other researchers which propose a model for extremely dense crowd counting based on Alexnet [65] pre-trained with ImageNet. They prove the beneficial of the pre-trained models for achieving better performance in the crowd counting. CNN-boosting [152] was one of the latest basic deep network for crowd counting. It proposes a new layer-wise manner to decline the training time and increase the counting accuracy. This model retains the benefit of selective sampling and layered boosting to achieve its goal.

**Multi-branch Architecture**: to address the scale variation in different respective fields, researchers have proposed CNN models based on Multi-branch networks,

which have shown to have boosted the performance and quality of density maps in crowd counting models.

Many studies have been done based on multi-branch architectures [138, 146]. One of the initial works was done by Zhang et al. [183], who proposed a three-CNN-column based MCNN structure, each with different receptive parameters to handle a range of different head sizes. Based on the idea of MCNN, a multi-branch patch-based model, Switch-CNN [123, 183] was proposed by Sam et al.. Their approach used the concept of patch classification and multi-scale regressors for generating the density map. CrowdNet [8] utilises a mixture of shallow and deep architectures with different structures for extracting low-level and semantic information within the images. Onoro et al. [107] introduce Hydra-CNN which proposes a novel pyramid patch-based solution based on a CNN regressor to handle the scale changes in the scene, and it produces more accurate density maps.

Furthermore, Sindagi et al. [136] combine various modules to capture the local and global information and generate density maps. In addition to these, it uses an adversarial learning [40] to fuse global, local and density estimator features from various levels. Sam et al. [120] proposed TDF-CNN which consists of three steps, i.e., a bottom-up CNN to generate the initial density map results, a top-down feedback generator to provide masks as feedback of contextual information, and finally a network similar to bottom-up to apply the feedback to the model. Liu et al.were the other researchers that utilise multi-branch structure for crowd counting. They proposed DRSAN [85] which addresses the rotation and scale variation within the scene by utilizing the Spatial Transformer Network (STN) [57]. Sindagi et al.proposed a new multi-branch network, i.e., CP-CNN [136], which added two other branches to classify an image-wise density to provide the global and local context information to the MCNN model. Deb et al. [24] incorporated the Atrous convolutions into the multi-branch network by assigning different dilation rates to various branches.

In 2019 many researchers proposed novel solutions based on multi-branch structure. Hossain et al. proposed SAAN [49], which attempts to capture local and global information in the same way as MoC-CNN [67] and CP-CNN [136]. However, SAAN takes the advantage of visual attention mechanism to automate the selection of the proper scale for the input image data. RANet was proposed by Zhang et al. to handel the issues of capturing long-range and short-range interdependence information with global self-attention (GSA) and local self-attention (LSA) respectively. Moreover, it utilises a relation module to combine the LSA and GSA to produce better representative feature maps. McML was introduced by [18] is another multi-branch network which employs a statistical model via the multi-branch architecture to approximate the relationship between various columns. This mutual learning idea optimizes each column alternately with fixing the weight of the other columns in every mini-batch training data. DADNet [42] captures the contextual and scale variation information by employing the dilated-CNN with various dilated rates. It also uses the deformable convolution to produce more accurate density maps which located each object feature maps accurately.

Recently, Tian et al. proposed PaDNet [146], which was composed of several components such as the Density-Aware Network (DAN), Feature Enhancement Layer (FEL), and a Feature Fusion Network (FFN). PaDNet improved the-state-of-the-art results remarkably by capturing pan-density information and utilizing global and local contextual features. IG-CNN [5] was another extensive study that combined the clustering and crowd counting for estimating the density map more adaptively based on training a mixture of experts that could incrementally adapt and grow based on the complexity of the dataset.

Although multi-branch networks have been improved the accuracy and performance of the crowd counting solutions, they still have some fundamental disadvantages. Li et al. [76] revealed the following issues through experiments. A) Training

of these models needs more time due to their multiple branches. B) They have a lot of redundant information since all the branches have similar structures. C) Multi-branch solutions always need to classify the density before passing images into the models, and due to the huge variation of the density within the input scene, defining the granularity of density level becomes problematic. D) Besides, having more fine-grained classifiers is equivalent to adding more layers and increasing the complexity of the structure of the model. Thus, due to all the demerits mentioned above, Multi-branch solutions are not proper for all scenarios. Therefore, many researchers have changed their focus toward more straightforward, efficient, and practical solutions. Consequently, Single-branch models have emerged to address the need for less complicated solutions for various circumstances in the crowd density estimation and counting task.

**Single-Branch Architecture**: aiming to develop a single branch and deeper convolutional networks to reduce the complexity of model compared with multi-columns network architecture. Their architectural simplicity and training performance help Single-branch approaches obtain popularity in recent years.

Sheng et al. proposed W-VLAD [129], which is a model based on spatial cues and semantic features. This model introduces a new locality-aware feature (LAF) for capturing the location-wise information within the scene. SaCNN was introduced by [179] in 2018, it employs a network scale-adaptively which consists of a backbone FCN. It combines different layers of feature maps to reflect the scale changes within the image and generate the final density map result. Another interesting research in single-branch crowd counting is D-ConvNet [132]. D-ConvNet augments the generalization power of ensemble solutions with negative correlation learning (NCL) and a group of weak regressors which are fed with convolutional feature maps.

One of the game changer in crowd counting domain is CSRNet [76] which com-

bines the VGG16 [134] pre-trained feature maps with the dilated convolution layers to enlarge the respective field and to address the scale variation effectively. After CSRNet, Cao et al.proposed SANet [9] which further improved CSRNet results. SANet deploys a model based on the Inception architecture [143] backbone. It uses the concept of multi-scale features encoder and decoder based on transposed convolution layers to address the scale variation challenges.

Pyramid modules were introduced by Zhao et al. [184] to produce proper quality features on the scene semantic segmentation task. They introduced an efficient method to estimate human head sizes and integrated them to an attention module to aggregate density maps from different layers and generate the final density map. SPN [15] is one of the first crowd counting model which takes the advantageous of the Scale Pyramid Module (SPM) to extract multi-scale features from a shared deep single column network. It also leverages the dilated layers with various dilation rates to produce a density map with better quality. Some of the recent studies focused on utilizing pyramid and attention-based modules [150]. Liu et al. [88] presented another end-to-end multi-scaled solution CAN based on fusing multi-scale pyramid features. They used modified PSP modules for extracting multi-scale features from the VGG16 [134] features to address the rapid scale changes within the scenes. Their model leveraged multi-scale adaptive pooling operations to cover a variety range of receptive fields. Compared to CAN, Chen et al. proposed an end-to-end single-column structure as a Scale Pyramid Network (SPN), which extracted multi-scale features with the dilated convolutions with various dilation rates $(2, 4, 8,$ and $12)$ from the VGG16 [134] backbone features [15]. The experimental results proved that their idea worked well on some well-known datasets.

On the other hand, the attention module and the idea proposed in [51] aimed to re-calibrate the features adaptively, so as to highlight the effect of valuable features, while suppressing the impact of weak ones [118]. Recently, researchers attempted to

incorporate this module and its variations into their single-branch models to improve the performance in several tasks such as object detection, object classification, and medical image processing [58, 125, 170]. Rahul et al. proposed an attention-based model to regress multi-scale density maps from several intermediate layers [86]. ADCrowdNet [86] was one of the latest researches in the area of crowd counting, and it used attention modules to generate accurate density maps. Liu et al. utilised a two-step cascade encoder-decoder architecture, one for the detection of crowded areas and producing the attention map as Attention Map Generator (AMG), and the other for generating the density map called Density Map Estimator (DME). Their method achieved excellent results on the ShanghaiTech Part A dataset.

Recently, W-Net [149] leverages the idea from U-Net [117], and adds an auxiliary reinforcement branch to ease the convergence. It takes advantage of the Structural Similarity Index (SSIM) to generate the output density. Varior et al.proposed SAA-Net [150] which incorporates a hierarchical CNNs structure by training of a group of attention module on the feature maps of middle layers. This idea is, to some extent, similar to SaCNN [179], the only difference between these two is the addition of attention masks. Recently, Jiang et al.introduced TEDnet [61], it mimics encoder-decoder CNN architecture., It handles the scale changes with an integration of multiple decoding paths and deploys a kind of skip connection to receive the supervised information. It also proposes a combination of local coherence and spatial correlation losses to solve the vanishing gradient issue.

## 2.3 Datasets

Nowadays, several crowd-counting datasets have been introduced due to the surge in popularity and the applications of crowd counting in public safety and crowd management. These datasets try to address various crowd counting challenges and from scale variations to the real-world applications' illumination variation. We

review most of the existing crowd counting datasets. Various tables in this chapters provide a list of existing crowd counting datasets with their various features such as their applications, the number of samples in each dataset, and image dimension. For the table of these section, we record the data from the [35], and due to the nature of datasets, we do not change the tables information

### 2.3.1 Popular Datasets

At first, we briefly review five popular datasets that have been used for benchmarking and reporting the results in most of crowd counting papers. We collect the dataset information from the Gao et al.review paper [35]. Table 2.2 provides the statistic information related to these popular datasets.

• **UCSD [11]** is the first crowd counting dataset, which contains 2,000 images with the size of 238×158 from a single fixed location sidewalk camera. The crowd count in this dataset are between 11 to 46. The perspective of all images is similar due to collecting data from a single location.

Table 2.2 : **The most popular crowd counting dataset [35].**

| | Number of Data | Count Statistics | | | | Resolution | Attributes | Year |
|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Avg | Total | | | |
| UCSD [11] | 2000 | 11 | 46 | 24.9 | 49,885 | 238 × 158 | Real-world | 2008 |
| Mall [14] | 2000 | 13 | 53 | 31 | 62,325 | 320 × 240 | Real-world | 2012 |
| UCF_CC_50 [55] | 50 | 94 | 4,543 | 1,280 | 63,974 | 2101 × 2888 | Real-world | 2013 |
| WorldExpo'10 [176] | 3980 | 1 | 253 | 50.2 | 199,923 | 576 × 720 | Real-world | 2015 |
| SHT_A [183] | 482 | 33 | 3,139 | 501.4 | 241,677 | 589 × 868 | Real-world | 2016 |
| SHT_B [183] | 716 | 9 | 578 | 123.6 | 88,488 | 768 × 1024 | Real-world | 2016 |

• **Mall [14]** is the second dataset which contains 2,000 image with the fixed size of 320×240, and it has a crowd variation from 13 to 53 with the total number of 62,325 in the dataset. This dataset has more density and perspective variation compared with UCSD [11]. Besides, it contains more scenes with the severe occlusions and activity patterns changes due to the static and moving persons in each scene.

• **UCF_CC_50 [55]** is really difficult and tough dataset, which contains only 50 images from various scenes collected and annotated from public web images. Compared with UCSD [11] and Mall [14], it has significantly more people variation from 94 to 4543, with the average and total number equal to 1,280 and 63,974, respectively. Due to the fewer numbers of images, usually researchers uses a 5-fold cross-validation for reporting the results. Due to the fewer scenes in this dataset, it is a tough dataset for crowd counting models (even the newest ones).

• **WorldExpo'10 [176]** is a large dataset with thousands of labelled video frames collected with street-view cameras from Shanghai World-Expo2010. This dataset contains 199,923 annotated people from 3,920 frames with an image size of 576×720 divided into five subsets.

• **Shanghai Tech [183]** is one of the most referenced dataset in the crowd counting task. This large-scale dataset includes 1,198 scenes with 330,165 ground truth labeled people. This dataset is consists of two different parts, Part_A(SHT_A) as high-density distributing data, and Part_B (SHT_B) as a low-density dataset. Part B contains the scenes from a busy street in Shanghai, whilst Part A includes internet-based randomly selected images. The perspective distortion and scale change presented by these two parts provide a real challenging benchmark for the new crowd counting CNN-based models.

### 2.3.2 Recently Proposed Datasets

In addition to the above datasets, recently researchers proposed some good benchmarking datasets such as Smartcity [179], UCF-QNRF [56],City Street [180], JHU-CROWD [141], NWPU-Crowd [155] and etc. Among these recently proposed datasets, UCF-QNRF [56], JHU-CROWD [141], and NWPU-Crowd [155] are more challenging ones with the huge number of annotated data and the large range of density variation as it is shown in Table 2.3. Thus, the recent crowd counting studies have begun to report their model performance on these datasets.

Table 2.3 : **The most recent introduced crowd counting dataset [35].**

| Dataset | Number of Data | Count Statistics | | | | Resolution | Attributes | Year |
|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Average | Total | | | |
| Smartcity [179] | 50 | 1 | 14 | 7.4 | 369 | 1920 × 1080 | Real-world | 2018 |
| UCF-QNRF [56] | 1,535 | 49 | 12,865 | 815 | 1,251,642 | 2013 × 2902 | Real-world | 2018 |
| Shanghai TechRGBD [77] | 2193 | 6 | 234 | 65.9 | 144,512 | 1080 × 1920 | Real-world | 2019 |
| Crowd Surveillance [168] | 13,945 | ** | ** | 35 | 386,513 | 1342 × 840 | Real-world | 2019 |
| FDST [29] | 15,000 | 9 | 57 | 26.7 | 394,081 | 1920 × 1080<br>1280 × 720 | Real-world | 2019 |
| City Street [180] | 500 | 70 | 150 | ** | ** | 676 × 380 | Real-world | 2019 |

**Table 2.3 continued from previous page**

| Dataset | Number of Data | Count Statistics | | | | Resolution | Attributes | Year |
|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Avg | Total | | | |
| JHU-CROWD [141] | 4250 | ** | 7286 | 262 | 1,114,785 | 1450 × 900 | Real-world | 2019 |
| Drone Crowd [161] | 33,600 | 25 | 455 | 144.8 | 4,864,280 | 1920 × 1080 | Drone-based | 2019 |
| GCC [156] | 15,212 | 0 | 3,995 | 501 | 7,625,843 | 1080 × 1920 | Synthetic | 2019 |
| DLR-ACD [6] | 33 | 285 | 24,368 | 6857 | 226,291 | 3619 × 5226 | Aerial imagery | 2019 |
| NWPU-Crowd [155] | 5,109 | 0 | 20,033 | 418 | 2,133,238 | 2311 × 3383 | Real-world | 2020 |

### 2.3.3 Special Crowd Counting Datasets

Besides the discussed datasets, there are many other domain specific datasets, which are only used in some particular crowd counting applications and scenarios. We list these datasets with their features in Table 2.4 [35]. These applications include train station crowd counting (STF [30] and TS [30]), subway station (Shanghai Subway Station [45]), crowd sequences (PETS [27], BRT (Beijing BRT [25]), airport (ZhengzhouAirport [60]), indoor (MICC [7], CIISR [167], and multi-sources (AHU-Crowd [78, 53].

Table 2.4 : **Special crowd counting datasets [35].**

| | Number of Data | Count Statistics | | | | Resolution | Attributes | Year |
|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Avg | Total | | | |
| PETS [27] | 1076 | 0 | 40 | ** | 18289 | 384×288 | Real-worl | 2010 |
| MICC [7] | 3358 | 0 | 28 | 5.25 | 17630 | ** | Real-world | 2014 |
| AHU-CROWD [53] | 107 | 58 | 2201 | ** | 45,000 | ** | Real-world | 2016 |
| Indoor[1] [95] | 570,000 | 0 | 59 | ** | ** | 704×576 | Real-world | 2016 |
| LHI[2] [186] | 3,100 | ** | ** | ** | 5,900 | 1280 × 720 | Real-world | 2016 |
| NWPU-Crowd [155] | 5,109 | 0 | 20,033 | 418 | 2,133,238 | 2311 × 3383 | Real-world | 2020 |
| Beijing BRT [25] | 1280 | 1 | 64 | ** | ** | 640 × 360 | Real-world | 2018 |
| Train Station [30] | 2000 | 1 | 53 | ** | 62581 | 256 × 256 | Real-world | 2017 |
| Shanghai Subway Station [45] | 3,000 | 28.78 | ** | ** | ** | ** | Real-world | 2017 |
| Indoor[2] [79] | 148,243 | 0 | 40 | 12.4 | 1,834,770 | 352 × 288<br>704×576 | Real-world | 2019 |
| Airport [60] | 1,111 | 7 | 128 | ** | 49,061 | ** | Real-world | 2019 |
| CIISR [167] | 1000 | ** | ** | 117 | ** | 1080 × 720 | Real-world | 2019 |

### 2.3.4 Representative Object Counting Datasets in Other Fields

Instance and object counting is the other application of crowd counting. Similarly, there are some datasets in other fields used to investigate the beneficial of CNN-based crowd density estimation models. For these purpose, some studies report their results based on dataset such as TRANCOS [41], WIDER FACE [169], Caltech [26] and etc. Table 2.5 provides information related to these datasets.

Table 2.5 : **Other counting datasets [35].**

| | Number of Data | Count Statistics | | | | Resolution | Attributes | Year |
|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Avg | Total | | | |
| Caltech [26] | 2000 | 6 | 14 | ** | 15043 | ** | Pedestrian detection | 2012 |
| TRANCOS [41] | 1244 | 9 | 107 | ** | 46,796 | 640 × 480 | Vehicle counting | 2015 |
| WIDER FACE [169] | 32,203 | ** | ** | ** | 393,703 | ** | Face detection | 2016 |
| Penguins [3] | 80095 | 0 | 67 | 7.18 | ** | ** | Penguins counting | 2016 |
| DukerMTMC [116] | 2 million | ** | ** | ** | 2,700 | 1920×1080 | tracking, human detection or ReID | 2016 |
| CARPK [50] | 1448 | ** | ** | ** | 89,777 | ** | Drone view-based car counting | 2017 |

**Table 2.5 continued from previous page**

| | Number of Data | Count Statistics | | | | Resolution | Attributes | Year |
|---|---|---|---|---|---|---|---|---|
| WebCamT [182] | 60 million | ** | ** | ** | ** | 352×240 | WebCam traffic counting | 2017 |
| MTC [93] | 361 | ** | ** | ** | ** | ** | Planting counting | 2017 |
| Wheat-Spike [62] | 20 | 749 | 1287 | 1005 | 20,101 | 1k∼3k | wheat spikes counting | 2018 |
| DCC [100] | 177 | 0 | 101 | 34.1 | ** | ** | Cell counting | 2018 |
| VisDrone2019 Veheicle [189] | 5303 | 10 | 349 | 37.52 | 198,984 | 991×1511 | Drone-based vehicle counting | 2018 |
| VisDrone2019 People [189] | 3347 | 10 | 289 | 32.41 | 108,464 | 969×1482 | Drone-based crowd counting | 2018 |

## 2.4 Evaluation Metrics

For every computer application, there is a need for defining some evaluation metrics to investigate the performance of the solutions. In crowd counting, we have several metrics to assess the performance of the model by comparing the predicted results with the annotated ground truths. In this section, we present two major measures for evaluating crowd counting models. Due to the aim of crowd counting,

we only focus on the image-level metrics for evaluation of the counting performance.

Two most common metrics in crowd counting are Mean Absolute Error (MAE) and the Root Mean Squared Error (MSE) [9, 76, 15, 88], which are defined by:

$$\text{MAE} = \frac{1}{M}\sum_{m=1}^{M}\left|C_m^{est} - C_m^{gt}\right|, \tag{2.1}$$

and

$$\text{MSE} = \sqrt{\frac{1}{M}\sum_{m=1}^{M}\left(C_m^{est} - C_m^{gt}\right)^2}, \tag{2.2}$$

where $M$ is the number of training or testing data, $C_m^{gt}$ denotes the exact number of people inside the ROI of the $m$-th scene and $C_m^{est}$ is the correspondingly estimated number of crowd [88].

Some benchmark datasets, such as UCSD [11], provide ROIs . If ROIs are not provided, the whole image is considered as an ROI. In this thesis, we apply the same rule as [9] to prepare the annotated information and ground truth data. Note that the number of people in an image can be calculated by the summation over the pixels of the ground truth ($D_i^{gt}$) and the predicted density maps ($D_i^{est}$). We follow the methodology in [9] to prepare the ground truth density data.

## 2.5 Evaluation and Discussion

This section provides the performance evaluation results for the recent deep learning based methods and seven representatives of typical models over five different mainstream benchmark crowd counting datasets. At first, we present the evaluation of models on the different benchmark datasets, and then determine the properties of the top three models based on MAE and MSE in each dataset. We report the reults on each dataset in a separate table, to be more understandable and clear. The information in the tables in this section are recalled from the published survay paper [35] of Gao et al.and the original published work. In order to highlight the

best models, we utilise three colors, red, green and blue, corresponding to the first best, the second best and the third best models, respectively.

### 2.5.1 Evaluation Results

Table 2.6 presents the results based on the ShanghaiTech Dataset, SHT_A and SHT_B [183] dataset, using the MSE and MAE metrics, which assess the robustness and accuracy of crowd counting solutions. As it is shown in this table, PCC Net [36] and SANet [9]+SPANet [17] are the best model on SHT_A and SHT_B, respectively. The SPANet [17] is a multi-branch model that utilises the pan-density/sub-region modules to estimate the number of crowd. On the other hand, PCC Net [36] is a single-branch model, which considers the perspective information to generate the output density map. The second best model is S-DCNet [164], which has a single-branch network consisting of a pyramid pooling to capture various density information.

Table 2.6 : **Evaluation of modern CNN-based crowd counting approaches on the ShanghaiTech Dataset, SHT_A and SHT_B [183] dataset. The results of this table collected from the table in [35].**

| Methods | PartA | | PartB | |
|---|---|---|---|---|
| | MSE | MAE | MSE | MAE |
| LBP+RR [14] | 371.0 | 303.2 | 81.7 | 59.1 |
| CP-CNN [136] | 106.4 | 73.6 | 30.1 | 20.1 |
| Switching CNN [123] | 135.0 | 90.4 | 33.4 | 21.6 |
| MCNN [183] | 173.2 | 110.2 | 41.3 | 26.4 |
| Cross-scene [176] | 277.7 | 181.8 | 49.8 | 32.0 |
| ACSCP [128] | 102.7 | 75.7 | 27.4 | 17.2 |
| DRSAN [85] | 96.4 | 69.3 | 18.2 | 11.1 |
| SaCNN [179] | 139.2 | 86.8 | 25.8 | 16.2 |
| DecideNet [81] | ** | ** | 31.98 | 21.53 |
| CSRNet [76] | 115.0 | 68.2 | 16.0 | 10.6 |

**Table 2.6 continued from previous page**

| Methods | PartA | | PartB | |
|---|---|---|---|---|
| | MAE | MSE | MAE | MSE |
| SANet [9] | 104.5 | 67.0 | 13.6 | 8.4 |
| SCNet [159] | 107.9 | 71.9 | 14.4 | 9.3 |
| ic-CNN (two stages) [111] | 116.2 | 68.5 | 16.0 | 10.7 |
| ASD [162] | 98.0 | 65.6 | 13.7 | 8.5 |
| PaDNet [146] | 98.1 | **59.2** | 12.2 | 8.1 |
| PACNN [130] | 106.4 | 66.3 | 13.5 | 8.0 |
| SAA-Net [150] | 104.1 | 63.7 | 12.7 | 8.2 |
| MA-Net [59] | 100.0 | 61.8 | 13.3 | 8.6 |
| CAN(ECAN) [88] | 100.0 | 62.3 | 12.2 | 7.8 |
| SFANet [188] | 99.3 | 59.8 | 10.9 | 6.9 |
| SFCN†[2] [156] | 107.5 | 64.8 | 13.0 | 7.6 |
| DUBNet [102] | 111.1 | 66.4 | 15.1 | 9.4 |
| CFF [131] | 109.4 | 65.2 | 12.2 | 7.2 |
| W-Net [149] | 97.3 | 59.5 | **10.3** | 6.9 |
| CTN [151] | 107.0 | 64.3 | 14.6 | 8.6 |
| SAAN [49] | ** | ** | 28.41 | 16.86 |
| DSNet [21] | 102.6 | 61.7 | 10.5 | **6.7** |
| DENet [83] | 101.2 | 65.5 | 15.4 | 9.6 |
| TEDnet [61] | 109.1 | 64.2 | 12.8 | 8.2 |
| ADCrowdNet [86] | 115.2 | 70.9 | 12.9 | 7.7 |
| RAZ-Net [80] | 106.7 | 65.1 | 14.1 | 8.4 |
| PSDDN [91] | 159.2 | 85.4 | 27.9 | 16.1 |
| RReg(CSRNet) [153] | 96.2 | 63.1 | 13.56 | 8.72 |
| SPN [15] | 99.5 | 61.7 | 14.4 | 9.4 |
| PCC Net [36] | 124.0 | 73.5 | 19.0 | 11.0 |
| DSSINet [84] | 96.04 | 60.63 | **10.34** | **6.85** |
| IA-DCCN [139] | 108.4 | 66.9 | 16.0 | 10.2 |
| SANet [9]+SPANet [17] | **92.5** | 59.4 | **9.9** | **6.5** |
| BL [97] | 101.8 | 62.8 | 12.7 | 7.7 |

**Table 2.6 continued from previous page**

| Methods | PartA | | PartB | |
|---|---|---|---|---|
| | MAE | MSE | MAE | MSE |
| HA-CCN [138] | 94.9 | 62.9 | 13.4 | 8.1 |
| L2SM [166] | 98.4 | 64.2 | 11.1 | 7.2 |
| MBTTBF-SCFB [140] | **94.1** | 60.2 | 15.5 | 8.0 |
| RANet [174] | 102.0 | 59.4 | 12.9 | 7.9 |
| PGCNet [168] | **86.0** | **57.0** | 13.7 | 8.8 |
| ACSPNet [96] | 137.1 | 85.2 | 23.1 | 15.4 |
| ANF [175] | 99.4 | 63.9 | 13.2 | 8.3 |
| LSC-CNN [121] | 117.0 | 66.4 | 12.7 | 8.1 |
| S-DCNet [164] | 95.0 | **58.3** | 10.7 | **6.7** |

The second dataset, which we investigate, is the UCF_CC_50 [55] dataset. Table 2.7 presents the results of Modern CNN-based model on this challenging dataset. DSNet [21] outperforms the other models using both metrics. As we look at the features of this model, we notice that this model is based on single-branch architectures and utilises the dilated convolution to provide a better respective field coverage. The second best models are CAN(ECAN) [88] and PaDNet [146]. CAN(ECAN) [88] is a single-branch model, which uses the dilated convolution and the pyramid pooling for extracting the context information from the input data. On the other hand, PaDNet [146] is a multi-branch model, which utilises the pyramid pooling and pan-density modules to deliver better density estimations.

Table 2.7 : **Evaluation of modern CNN-based crowd counting approaches on the UCF_CC_50 [55] dataset. The results of this table collected from the table in [35].**

| Methods | UCF [55] | |
|---|---|---|
| | MAE | MSE |
| Lempitsky et.al [71] | 487.1 | 493.4 |
| Idrees 2013 [55] | 590.3 | 468.0 |
| CP-CNN [136] | 320.9 | 295.8 |
| Switching CNN [123] | 439.2 | 318.1 |
| MCNN [183] | 509.1 | 377.6 |
| Cross-scene [176] | 498.5 | 467.0 |
| ACSCP [128] | 404.6 | 291.0 |
| DRSAN [85] | **250.2** | 219.2 |
| SaCNN [179] | 424.8 | 314.9 |
| CSRNet [76] | 397.5 | 266.1 |
| SANet [9] | 334.9 | 258.4 |
| SCNet [159] | 332.8 | 280.5 |
| ic-CNN (two stages) [111] | 365.5 | 260.9 |
| ASD [162] | 270.9 | 196.2 |
| PaDNet [146] | 278.3 | **185.8** |
| PACNN [130] | 357.8 | 267.9 |
| SAA-Net [150] | 310.8 | 238.2 |
| MA-Net [59] | 349.3 | 245.4 |
| CAN(ECAN) [88] | **243.7** | 212.2 |
| SFANet [188] | 316.2 | 219.6 |
| SFCN†[2] [156] | 318.2 | 214.2 |
| DUBNet [102] | 332.7 | 235.2 |
| W-Net [149] | 309.2 | 201.9 |
| CTN [151] | 331.0 | 219.3 |
| SAAN [49] | 391.0 | 271.6 |
| DSNet [21] | **240.6** | **183.3** |

**Table 2.7 continued from previous page**

| Methods | UCF [55] | |
|---|---|---|
| | MAE | MSE |
| DENet [83] | 345.4 | 241.9 |
| TEDnet [61] | 354.5 | 249.4 |
| ADCrowdNet [86] | 362.0 | 273.6 |
| PSDDN [91] | 514.8 | 359.4 |
| SPN [15] | 335.9 | 259.2 |
| PCC Net [36] | 315.5 | 240.0 |
| DSSINet [84] | 302.4 | 216.9 |
| IA-DCCN [139] | 394.4 | 264.2 |
| SANet [9]+SPANet [17] | 311.7 | 232.6 |
| BL [97] | 308.2 | 229.3 |
| HA-CCN [138] | 348.4 | 256.2 |
| L2SM [166] | 315.3 | **188.4** |
| MBTTBF-SCFB [140] | 300.9 | 233.1 |
| RANet [174] | 319.4 | 239.8 |
| PGCNet [168] | 317.6 | 259.4 |
| ACSPNet [96] | 383.7 | 275.2 |
| ANF [175] | 340.0 | 250.2 |
| LSC-CNN [121] | 302.7 | 225.6 |
| S-DCNet [164] | 301.3 | 204.2 |

The third dataset, that we investigate, is UCF-QNRF [56]. Table 2.8 provides the CNN-based model evaluation results on this dataset. The results show that BL [97] has the best accuracy on this dataset. Interestingly, BL is a single-branch model with any other fascinating feature, and it proves the power of single-branch structures. The second best model is DSNet [21], which further explains why the single-branch model becomes popular nowadays.

Table 2.8 : **Evaluation of modern CNN-based crowd counting approaches on the UCF-QNRF [56] dataset. The results of this table collected from the table in [35].**

| Methods | UCF-QNRF [56] | |
|---|---|---|
| | MAE | MSE |
| Idrees 2013 [55] | 508.0 | 315.0 |
| Switching CNN [123] | 445 | 228 |
| CSRNet [76] | 208.5 | 120.3 |
| PaDNet [146] | 170.2 | **96.5** |
| SAA-Net [150] | 97.5 | 167.8 |
| CAN(ECAN) [88] | 183 | 107 |
| SFANet [188] | 174.5 | 100.8 |
| SFCN†[2] [156] | 171.4 | 102.0 |
| DUBNet [102] | 178 | 116 |
| DSNet [21] | **160.4** | **91.4** |
| TEDnet [61] | 188 | 113 |
| RAZ-Net [80] | 195 | 116 |
| PCC Net [36] | 191 | 132 |
| DSSINet [84] | **159.2** | 99.1 |
| IA-DCCN [139] | 185.7 | 125.3 |
| BL [97] | **154.8** | **88.7** |
| HA-CCN [138] | 180.4 | 118.1 |
| L2SM [166] | 173.6 | 104.7 |
| MBTTBF-SCFB [140] | 165.2 | 97.5 |
| RANet [174] | 190 | 111 |
| ANF [175] | 174 | 110 |
| LSC-CNN [121] | 218.2 | 120.5 |
| S-DCNet [164] | 176.1 | 104.4 |

The fourth dataset, that we investigate, is WorldExpo'10 [176], which consists of five different sections. Researchers generally report the average of five sections. Table 2.9 presents the evaluation results on this dataset. The results show that SCNet [159] has the best performance in this dataset. SCNet is a single-branch model, which consists of dilated convolution and pyramid pooling modules. As we notice, the proper combination of these three features easily provides a model with a good performance in the crowd counting. The second best model is DSSINet [84], which has a multi-branch network. It delivers good results with the help of dilated convolution and attention modules. The third best model on this dataset is CAN(ECAN) [88].

Table 2.9 : **Evaluation of modern CNN-based crowd counting approaches on the WorldExpo'10 [176] dataset. The results of this table collected from the table in [35].**

| Methods | WorldExpo'10 [176] | |
|---|---|---|
| | MAE | MSE |
| LBP+RR [14] | ** | 31.0 |
| CP-CNN [136] | ** | 8.86 |
| Switching CNN [123] | ** | 9.4 |
| MCNN [183] | ** | 11.6 |
| Cross-scene [176] | ** | 12.9 |
| ACSCP [128] | ** | 7.5 |
| DRSAN [85] | ** | 7.76 |
| SaCNN [179] | ** | 8.5 |
| DecideNet [81] | ** | 9.23 |
| CSRNet [76] | ** | 8.6 |
| SANet [9] | ** | 8.2 |
| SCNet [159] | ** | **6.4** |
| ic-CNN (two stages) [111] | ** | 10.3 |
| PACNN [130] | ** | 7.8 |

**Table 2.9 continued from previous page**

| Methods | WorldExpo'10 [176] | |
|---|---|---|
| | MAE | MSE |
| MA-Net [59] | ** | 8.34 |
| CAN(ECAN) [88] | ** | 7.4 (**7.2**) |
| DENet [83] | ** | 8.2 |
| TEDnet [61] | ** | 8.0 |
| ADCrowdNet [86] | ** | 7.3 |
| RAZ-Net [80] | ** | 8.0 |
| PSDDN [91] | ** | ** |
| AT-CSRNet [185] | ** | 7.8 |
| RReg(CSRNet) [153] | 8.5 | ** |
| PCC Net [36] | ** | 9.5 |
| DSSINet [84] | ** | **6.67** |
| SANet [9]+SPANet [17] | ** | 7.7 |
| PGCNet [168] | ** | 8.1 |
| ACSPNet [96] | ** | 9.8 |
| ANF [175] | ** | 8.1 |
| LSC-CNN [121] | ** | 8.0 |

Finally, we present the evaluation results on the UCSD [11] dataset. UCSD is a low crowded dataset with a little variation in the density. Generally, researchers consider this dataset as a sparse crowd counting dataset. Table 2.10 shows the performance results of modern crowd counting methods. As shown, W-Net [149], besides DSNet [21] and SFANet [188], provides the best accuracy on this dataset. After looking at their characteristics, we notice that all of these methods are single-branch models, which employ the other features such as attention module and dilated convolution to produce the most accurate density maps.

Table 2.10 : **Evaluation of modern CNN-based crowd counting approaches on the UCSD [11] dataset. The results of this table collected from the table in [35].**

| Methods | WorldExpo'10 [176] | |
|---|---|---|
| | MAE | MSE |
| Lempitsky et.al [71] | ** | 1.70 |
| GP [11] | 7.97 | 2.24 |
| Switching CNN [123] | 2.10 | 1.62 |
| MCNN [183] | 1.35 | 1.07 |
| Cross-scene [176] | 3.31 | 1.60 |
| ACSCP [128] | 1.35 | 1.04 |
| CSRNet [76] | 1.47 | 1.16 |
| SANet [9] | 1.29 | 1.02 |
| PaDNet [146] | **1.06** | **0.85** |
| PACNN [130] | ** | **0.89** |
| SFANet [188] | **1.07** | **0.82** |
| DUBNet [102] | 1.24 | 1.03 |
| W-Net [149] | **1.05** | **0.82** |
| DSNet [21] | **1.06** | **0.82** |
| DENet [83] | 1.31 | 1.05 |
| ADCrowdNet [86] | 1.35 | 1.09 |
| SPN [15] | 1.32 | 1.03 |
| SANet [9]+SPANet [17] | 1.28 | 1.00 |
| ACSPNet [96] | 1.28 | 1.02 |

We can summarise the results shown in Tables 2.6-2.10 as follows.

• **Traditional models vs CNN-based models.** By looking at the Table 2.6 to Table 2.10, we can easily find that the CNN-based approaches outperform the traditional crowd counting approaches. It also proves the significant effect of convolutional feature maps in the crowd density estimation task.

• **Performance comparison of CNN-based models.** As shown in Tables 2.6 to Table 2.10, the accuracy and robustness of the CNN-based crowd counting models are significantly improved over the years. Among all CNN-based models, cross scene [176] presents the worst performance, and it is the first model that utilises CNNs for crowd counting.

### 2.5.2 Key Features of the Best Solutions

After reviewing the performance of modern CNN-based approaches on the five most used datasets, we can have a better understanding about characteristics of well-established crowd counting solution. We assess the properties of the CNN-based models, we select the top three solutions in term of RMSE and MAE metrics, over each of five common datasets. Besides, we add MCNN [183] and CSRNet [76] as two game-changing and heuristic models to collect the final 19 models. The main characteristics of these 19 models illustrate in the Table 2.11. These characteristics help us to understand and explain the great performance of these state-of-the-art models.

Table 2.11 : **Main properties of the best crowd counting solutions [35].**

| Methods \ Properties | Multi-branch | Single-branch | Attention-based | Dilation convolution | Spatial transformer | CRFs/MRF | Perspective information | Pyramid pooling | Pan-density /sub-region |
|---|---|---|---|---|---|---|---|---|---|
| MCNN [183] | ✓ | | | | | | ✓ | | ✓ |
| DecideNet [81] | ✓ | | ✓ | | | | | | |
| CSRNet [76] | | ✓ | | ✓ | | | | | |
| SCNet [159] | | ✓ | | ✓ | | | | ✓ | |
| DRSAN [85] | ✓ | | ✓ | | ✓ | | | | |
| PACNN [130] | | ✓ | | | | | ✓ | | |
| CAN&ECAN [88] | | ✓ | | ✓ | | | ✓ | ✓ | |
| PaDNet [146] | ✓ | | | | | | | ✓ | ✓ |
| SFANet [188] | | ✓ | ✓ | | | | | | |
| SAAN [49] | ✓ | | ✓ | | | | | | |
| DSNet [21] | | ✓ | | ✓ | | | | | |
| SPANet [17] | ✓ | | | | | | | | ✓ |
| DSSINet [84] | ✓ | | ✓ | ✓ | | ✓ | | | |
| L2SM [166] | | ✓ | | | | | | | |
| PGCNet [168] | | ✓ | | | | | ✓ | | |
| W-Net [149] | | ✓ | ✓ | | | | | | |
| S-DCNet [164] | | ✓ | | | | | | | ✓ |
| MBTTBF-SCFB [140] | ✓ | | ✓ | | | ✓ | | | |
| BL [97] | | ✓ | | | | | | | |

Based on an initial investigation, it is clear that two-thirds of the best models apply the single column CNN network architecture. Thus, based on this fact, we can conclude that, instead of proposing wider multi-columns network solutions, it

is better to make a deeper network. Besides, among these state-of-the-art methods, we can figure out that more than one third of them take the advantageous of attention modules [85, 81, 86, 49, 150, 188, 131, 138] and dilated Convolutional layers [76, 162, 15, 86, 156, 88, 21]. In contrast with the traditional convolution modules, the attention module uses feature map information to compute the neural responses. Then, it utilises this information to assign weights to each pixel and channel in the feature maps. This procedure assists the model to focus on the important spacial and channel-based feature map information. Thus, CNN-based model can learn better in a more efficient way by distinguishing the critically important features. Moreover, it can accelerate convergence speed. Due to this remarkable power, attention mechanism became popular in various computer vision applications, for instance, visual pose estimation [20],image deblurring [110], semantic segmentation [114], and image classification [51]. Attention module also provides a good solution to create an ROI and masking out the irrelevant data and information. The dilated layer can enlarge the cover area by convolution without expanding parameters of the model or losing information caused by max and average pooling operation. Thus, the dilated convolution can be a good solution for capturing multi-scale features and maintaining more detailed information [35].

Deformable convolution [22] and spatial Transformer Network (STN) [57] are the other mechanisms to handle the scaling or warping, and rotation, which have some negative effect on the performance of standard convolutional layers. Distinctively, STN is a well-established mechanism that assists a CNN model in learning the spatial transformation between different data without further annotation data. It can perform the spatial transformation on any layer in the CNN-based model (input layer or middle layers ) to learn the spatial alteration of various feature maps. because of this prominent effect, STN has been used for several applications, e.g. saliency detection [66] and multi-label image recognition [160]. In the crowd counting

task, Liu et al. [85] used STN to manage the rotation and scale variation challenges.

Markov Random Fields (MRFs) [75] and Conditional random fields (CRFs) [68] are the other techniques which are used to do a post-processing operation on the output of the CNNs to improve the performance of the model with a message passing mechanism [64]. Liu et al. [84] applied CRFs on CNNs to refine the feature maps with several different scales, and proved the effectiveness of this mechanism on the benchmark crowd counting datasets. In the work of [175], they presented an attentional neural field (ANF) framework which uses CRFs for crowd counting task.

As we mentioned in the crowd counting challenges section 1.2 of Chapter 1, perspective distortion is one of the crucial challenges in the crowd counting. There are two ways to have the perspective information, i.e., using the camera's six-degree-of-freedom (DOF) [37], or assessing the scale variation by considering the distance from the camera. Many traditional crowd counting methods [11, 13] use the second method to provide the scale variation and perspective geometry information to normalize the regression features maps. Some researchers refine the body part maps [54] or the ground truth density map [176, 183] by utilizing the perspective information. Some recent works [88, 130] use the perspective information to transform local/global scales in the CNNs.

Pan-density crowd counting addresses the two critical issues in crowd scenarios, i.e., varying distributions and densities in various scenes and density variation within the same scene. Most of the Multi-branch models have a section to tackle this issue, such as CP-CNN [136], Switch-CNN [123] and MCNN [183], but as we mentioned previously, these models have several other issues such as high computation complexity, low efficiency. However, recently PaDNet [146] as a single column architecture provides a sub-module (Density-Aware Network (DAN)), to effectively identify local crowd and uses the Feature Enhancement Layer (FEL) to refine the

feature maps.

Spatial pyramid pooling (SPP) [47] is the other interesting module in the crowd counting area, which utilises the pooling to extract the multi-scale feature maps and then aggregate the outputs into a fixed sized feature map to boost the performance of crowd counting models. Besides, it can improve the convergence speed. Many modern crowds counting solutions SCNet [159], PaDNet [146] and CAN [88] utilise a form of SPP to incorporate the context information in their models and improve the accuracy of the crowd counting solutions.

We can summarise the Table 2.11 as follows.

• Single column CNN-based models are the dominant architecture, due to their efficiency in terms of performance and complexity as proved in [76].

• Utilising dilation convolution, spatial pyramid pooling (SPP), and the visual attention mechanism significantly elevates the performance of the crowd counting models in terms of accuracy and quality of density maps.

• Incorporating perspective information [176, 183, 88, 130] into the models can have a positive effect on addressing the scale variation challenges in the crowd counting task.

• Incorporating deformable convolution [22, 86] and spatial transformer network [57, 85] into the model can handle the rotation and uniform distributions of crowds.

• Pan-density learning [146] can address the local and global density variation problems.

• Multi-pathway or multi-task framework [81, 162, 188] with jointly loss function can accelerate convergence speed and improve the crowd counting performance.

# Chapter 3

# A-CCNN: Adaptive CCNN for Density Estimation and Crowd Counting

In this chapter, we propose an Adaptive Counting Convolutional Neural Network (A-CCNN) and consider the scale variation of objects in a frame adaptively so as to improve the accuracy of counting. Our proposed model first estimates the size of people's heads, and then utilises a fuzzy engine to determine the proper hyper parameter (patch size) needed for density prediction in each part of the image.

Our method takes advantages of contextual information to provide more accurate and adaptive density maps and crowd counting in the scene. Extensively experimental evaluation is conducted using different benchmark datasets for object-counting and shows that the proposed approach is effective and outperforms state-of-the-art approaches.

## 3.1 Introduction

Nowadays, estimating the number of people in a crowded scene is a desirable application especially in restricted, public event places such as train stations. Incidents, traffic delay and even terrible stampedes may be caused by overcrowding in such a scene. Generally, there is an urgent need for real-time decision making corresponding to crowd changes. To deal with this situation, there exist various challenges caused by occlusions, size and shape variations of people, perspective distortion, etc. Thus, correctly counting in crowded areas is very necessary in various real world tasks including traffic monitoring, visual surveillance, and crowd analysis.

The existing approaches can be classified into two main groups, i.e., detection based solutions and feature regression based solutions [137]. Detection based methods (also called direct methods) segment and detect every individual person or object in a scene with pre-trained classifiers and then simply count them. However, in some more complex scenes with serious occlusions and extremely crowded scenes, these approaches often fail to detect individuals and therefore produce inaccurate counting. In the feature regression based approaches (also called indirect approaches), learning algorithms or statistical methods are utilised to analyze the image appearance features of crowded scenes, and then estimate the number of people or objects based on image appearance. Thus, these methods are more suitable for dealing with highly crowded scenes where detecting individuals often fails.

In this chapter, based on the recent advance of Counting Convolutional Neural Network (CCNN) [107], we propose a new adaptive CCNN architecture, abbreviated as A-CCNN, that processes each part of an input image using the most appropriate trained CCNN model in order to estimate a density map more accurately. As illustrated in Fig. 3.1, to tackle the counting problem, our A-CCNN model is able to regress the density function corresponding to a specified section. This allows our model to accurately localize density map for unseen images. The most detectable features that make this model outstanding for crowd analysis are: (1) its capacity to handle scale variations in people's size when appearing in images, and (2) the facility to estimate local density in a crowdedness input image. Therefore, the proposed model can give a complete view about the scattering of the crowd. Compared to the prior studies, our solution does not use various CCNN networks, and only tries to choice the most proper Hyper Parameters (HPs) for generating a CCNN model. Thus, it can simply learn to address scale variations in an image with an effective way.

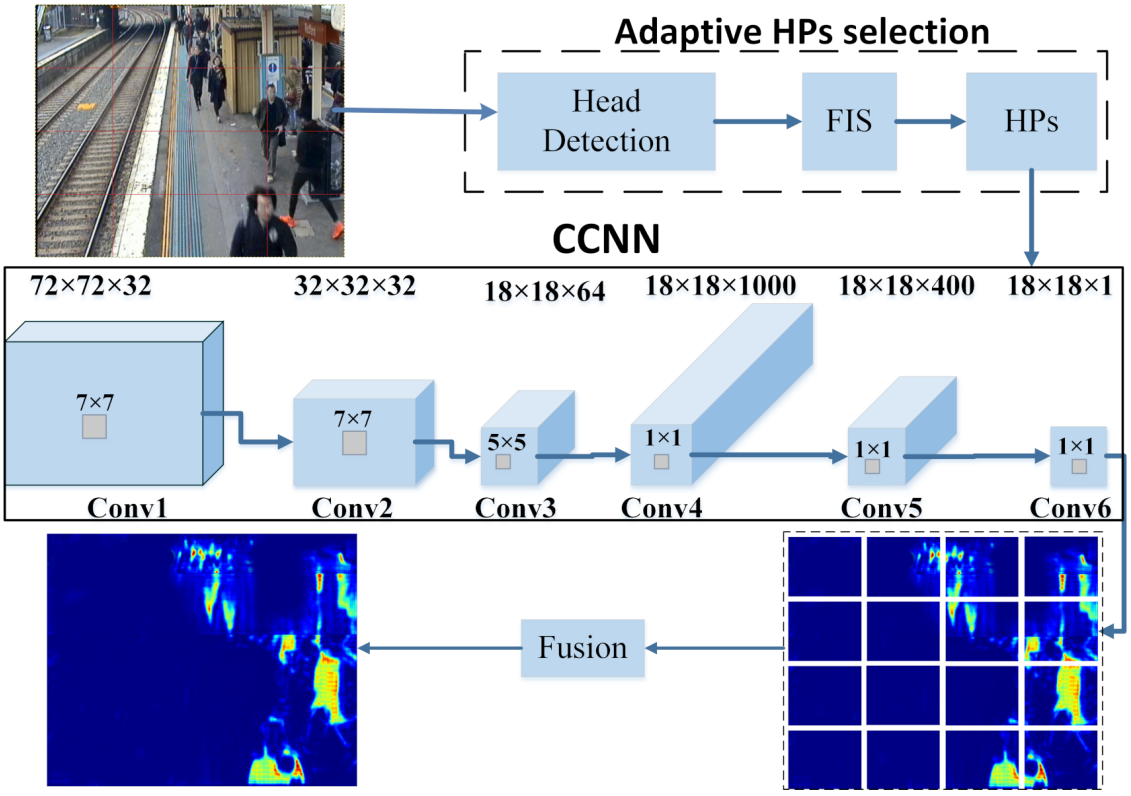



Figure 3.1 : **The overview of our proposed A-CCNN crowd counting method. For an input image, our A-CCNN first estimates head size and corresponding position, and then utilises a fuzzy engine to apply separate CCNN models to each section to estimate the overall count.**

### 3.1.1 The CCNN Architecture

The CCNN approach [107] takes a small patch of the image as an input and generates the corresponding density map for the image patch. By utilising the sliding window technique, it extracts patches and applies six layers of convolutional neural network to regress the density function. Therefore, the CCNN is formulated as a regression model that the network learns to generate object density maps based on the corresponding appearance of the image patches.

In the original CCNN model, the authors defined the ground truth density map $D_I$ as,

$$D_I(p) = \sum_{\mu \epsilon A_I} N(p, \mu, \Sigma). \tag{3.1}$$

where $A_I$ represents the number of annotated points in the input data $I$, and $N$ ($p$; $\mu$; $\Sigma$) represents a normalized 2D Gaussian function with a mean of $\mu$ and a covariance of $\Sigma$, evaluated at each pixel position $p$ [107].

The CCNN utilises two important HPs for generating models, i.e., the patch size and the value of $\Sigma$ in the Gaussian function. Through careful analysis of CCNN, we have noticed that it has a major problem in localizing the correct density map, in the way that it treats the whole parts of the input image in the same way. Therefore, it cannot achieve an acceptable accuracy in estimation when the scene has a large scale variation in the size of the objects. We observe that more accurate density maps can be produced, if we adaptively choose most proper values for the above two mentioned HPs.

## 3.2 Adaptive CCNN

In our work, in order to handle crowd images with large varieties in targets' appearance, we propose a new A-CCNN models for crowd counting. Our method adaptively apply CCNN models learned with different patch sizes for different regions according to subjects' sizes in an image/video.

As shown in Fig. 3.1, our A-CCNN architecture takes an image as the input and then determines the head size and position in different part of image. Then, by utilising a Fuzzy Inference System (FIS), it feeds each image section with the same FIS linguistic output value to an appropriate CCNN model with the proper HP to obtain the corresponding density map for each sections. In the end, it merges the output of different parts to obtain the final density map output.

In a detailed comparison of both the A-CCNN and the CCNN, we can discover

the following differences. First, we have utilised different sigma values to generate the training patches. The $\Sigma$ of the Gaussian function in Eq. 3.1 is changed to force the CCNN model to have a more accurate prediction about the number of crowd. Secondly, we have used distinct patch sizes for each part of the input image, instead of the same patch size for the whole image in the original CCNN.

During the experiment, we understand that if we increase the patch size the predicted number of crowd will decrease. In reality, the people who are near the camera are bigger than that of the people who are further. Thus, we use the lower sigma and patch size for the upper part and greater patch size for lower part of the image. By these two modifications, we have several separately trained CCNN models that can produce more accurate density maps, and then they can localize better the number of a targets in unseen scenes. Therefore, in testing time, the proposed model extracts sliding patches from the input image, and produces their corresponding density by utilising the relative CCNN model, then the density maps of these patches are assembled into the density map for each part [107]. Finally, the output density maps of each part are concatenated to produce the density map for the whole image.

In the proposed method, first, we perform tiny face detection [52] to estimate the sizes of heads in each part of the image. Then, by feeding the head sizes and the corresponding head positions to the FIS, we generate the appropriate HPs for the CCNNs. Finally, these HPs are used to produce an effective CCNN that can adaptively generate the final output density map.

### 3.2.1 Head Detection

To select the most suitable value for HPs, we need to know about the sizes of people or objects in different parts of an image. The tiny face detection approach [52] is used to detect some faces in each part of the input image. It creates a coarse image

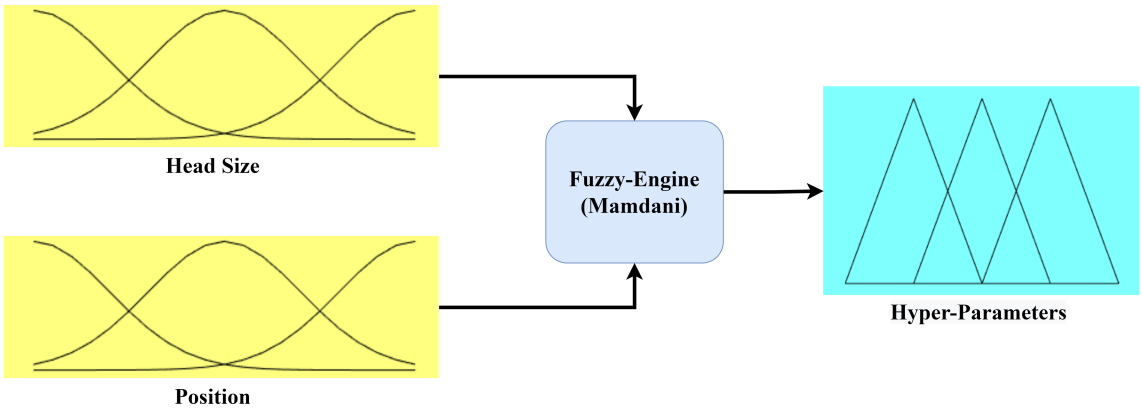



Figure 3.2 : **The fuzzy inference engine, where the head size and corresponding position are the two inputs and the level of HPs for CCNN is the output.**

pyramid of the input image, and then feeds the scaled inputs into the CNN to get the template responses. Finally, the final detection results are produced by applying the non-maximum suppression (NMS) at the original resolution. By applying this approach, we are able to obtain the size of the head and its position in each part of image.

### 3.2.2 Adaptive HP Selection by FIS

As shown in Fig. 3.1, in the next stage, we need to make a decision about the values of the HPs. Thus, the FIS is designed to adaptively select the value of the HPs by considering the size and the position of head. Fig. 3.2 shows the structure of FIS, as it is shown in this figure, it receives the fuzzy information about head size and position, and outputs in the form of fuzzy with the fuzzy linguistic variables. We choose the Gaussian membership function for all input and output variable. Small, Average, and Big are the fuzzy linguistic variables for head size and Up, Middle, and Down are the fuzzy linguistic values for head position, and the output linguistic variables are High-Pred, Mid-Pred, and Low-Pred.

Table 3.1 : **The fuzzy rule table for selecting HPs**

| Input | | Output |
|---|---|---|
| Head Size | Position | |
| Small | Up | High-Pred |
| Small | Middle | High-Pred |
| Small | Down | Mid-Pred |
| Average | Middle | Mid-Pred |
| Average | Down | Mid-Pred |
| Average | Up | Low-Pred |
| Big | Up | Mid-Pred |
| Big | Down | Low-Pred |
| Big | Middle | Low-Pred |

Based on the given membership function, the crisp input values are converted into a fuzzy linguistic variable based on the given membership function in FIS. Then, the fuzzy if-then rules developed based on Mamdani method [99] are used to map the input variables to appropriate fuzzy output variables. In total, nine fuzzy if-then rules are presented in Table 3.1. In general, the higher value of $\Sigma$ and sliding window (patch size) produce the density map with lower count of number of people, and high crowd counting is the output of lower amount of sigma and patch size. On the other hand, if the output of FIS is High-Pred, it means that we should use a CCNN that is trained with lower HP values.

### 3.2.3 Training Parameters

To demonstrate the effectiveness of the initial idea of adaptive CCNN architecture, we use the same parameters and function as in [107], for counting and

estimation. In our experiments, each CCNN is trained to regress the density map of the corresponding region of size $72 \times 72$. Similar to [107], the stochastic gradient decent algorithm is used during training. The loss weights that we used to balance the local count and global count are set to 0.9 and 0.1, respectively. The momentum and the learning rate are set to 0.9 and 0.01 respectively. We decrease the learning rate by a factor of 0.8 in each epoch. After 25 epochs, the model can reach a better local optimum. On the other hand, we utilise a small $\Sigma$ value and a small patch size for an area with a small head size, and a large $\Sigma$ value and a big patch size for an area with a big head size.

## 3.3 Experimental Results

To evaluate the performance of our A-CCNN algorithm, experiments are conducted on three different challenging crowd counting datasets, i.e., the UCSD dataset [11], the UCF-CC dataset [109], and the dataset of Sydney Train Station [30]. Note that the first two are public benchmark datasets.

The MAE is used as the evaluation metric for comparing the performance of A-CCNN against the state-of-the-art methods [107], and it is defined by Eq. 2.1.

### 3.3.1 The UCSD Dataset

We split the UCSD Dataset into four different subsets of training and testing images in the same way as in [107].

Table 3.2 presents the MAE results for our proposed A-CCNN and other state-of-the-art methods. As it shows, our A-CCNN performs competitively against other approaches with an MAE of 1.05 for the upscale subsets. Furthermore, in most of the other subsets, the results indicate that our proposed method has better performance compared with the original CCNN models especially in downscale one. However, the UCSD dataset is characterized by low variability in the scale of subjects and the

Table 3.2 : **Comparison of the MAE and MSE results obtained with our A-CNN and state-of-the-art crowd counting methods on UCSD crowd-counting dataset [11], UCF CC 50 (UCF) dataset [55], and Sydney Train Footage (STF) [30].**

| Method | MAE | | | | MSE | MAE | |
|---|---|---|---|---|---|---|---|
| | **UCSD** | | | | **UCF** | **STF** | |
| | **max** | **down** | **up** | **min** | | **C5** | **C9** |
| Density Learning [71] | 1.70 | 1.28 | 1.59 | 2.02 | 493.4 | - | - |
| Learning to Count[31] | 1.70 | 2.16 | 1.61 | 2.20 | - | - | - |
| Count Forest [109] | 1.43 | 1.30 | 1.59 | 1.62 | - | - | - |
| Arteta et al.[4] | 1.24 | 1.31 | 1.69 | 1.49 | - | - | - |
| Idrees et al.[55] | - | - | - | - | 419.5 | - | - |
| Zhang et al.[176] | 1.70 | 1.26 | 1.59 | 1.52 | 467.0 | - | - |
| Switch-CNN [123] | 1.65 | 1.79 | 1.11 | 1.50 | **318.1** | - | - |
| CrowdNet [8] | - | - | - | - | 452.5 | - | - |
| MCNN [183] | - | - | - | **1.35** | 377.6 | - | - |
| Farhood et al.[30] | - | - | - | - | - | 2.28 | 2.67 |
| HYDRA-CCNN [107] | - | - | - | - | 333.73 | - | - |
| CCNN [107] | - | - | - | 1.62 | 488.67 | 3.90 | 4.23 |
| **A-CCNN** | 1.63 | 1.60 | **1.05** | 1.61 | 375.2 | **1.69** | **1.87** |

crowd density in each frame. This limits the improvement achieved by our A-CCNN from leveraging intra-scene people scale variation.

### 3.3.2 The UCF-CC Dataset

We followed the same experimental settings as those of other state-of-the-art models [123] for the UCF CC 50 [55]. In Table 3.2, the MSE performance of our A-CCNN compared with other methods is shown. As it can be seen, our A-CCNN outperforms all other methods and demonstrates a 24 percentage improvement in MSE score compared to the CCNN, indicating the robustness of the predicted count to scenes with large variations of crowdness.

### 3.3.3 The Sydney Train Dataset

To evaluate the robustness of our model on real-world problems with heavy occlusions, low resolution and large variance in people's size, we have utilised the CCTV footage of a Sydney Train station and created annotated data for training and testing with our proposed approach. An example is included in Fig. 3.1. This dataset has two separate scenes and cameras C5 and C9 with 788 and 600 frames, respectively. The size of the input frames are $576\times704$, and the mask and annotation are provided. The huge variation in people's size and heavy extreme occlusions makes it be a very challenging task. Generally, in this dataset, the size of people who are in front of the cameras is three to four times larger than the people of further area. In this experiment, the input image is partitioned into three sections and three separate CCNN models are trained based on adaptive HPs.

Table 3.2 reports the MAE performance obtained on this dataset. The crowd count of A-CCNN is significantly higher than the original CCNN. This reinforces the fact that utilising our A-CCNN can efficiently manage both the difference in appearance and size of people. The various trained CCNNs employed by A-CCNN are able to provide a precise density map, independent of the dataset.

## 3.4 Conclusion

Aiming to tackle the difficult problem of crowd counting such as scale variance and extreme collusion, we present an Adaptive CCNN architecture that takes a whole image as input and directly outputs the density map. The proposed method makes a full use of contextual information to generate a more accurate density map. To leverage the local information, we have utilised the combination of CNN based head detection and fuzzy inference engine to choose the most suitable CCNN model adaptively to each part of the input image. We have achieved noticeable improvements on three challenging datasets, i.e., the UCSD, UCF-CC and the crowd dataset collected by ourselves from Sydney Trained Station, and demonstrates the effectiveness of the proposed approach.

# Chapter 4

# Performance-Enhancing Network Pruning for Crowd Counting

The Counting Convolutional Neural Network (CCNN) has been widely used for crowd counting. However, they typically end up with a complicated network model resulting in a challenge for real-time processing. The existing solutions aim to reduce the size of the network model, but unavoidably sacrifice the network accuracy. Different from the existing pruning solutions, in this chapter, a new pruning strategy is proposed by considering the contributions of various filters to the final result. The filters in the original CCNN model are grouped into positive, negative and irrelevant types. We prune the irrelevant filters of which feature maps contain little information, and the negative filters determined by a mask learned from the training dataset. Our solution improves the results of the counting model without fine-tuning or retraining the pruned model. We demonstrate the advantages of our proposed approach on the problem of crowd counting. Our experimental results on benchmark datasets show that the network model pruned using our approach not only reduces the network size but also improves the counting accuracy by 4% to 17% less MAE than the state-of-the-arts.

## 4.1 Introduction

The existing approaches for crowd counting can be roughly grouped into detection-based and feature-regression-based approaches. The detection-based approaches employ object detectors to detect or localize each person in the scene, and the counting is simply the number of total detection. These approaches can surpass

human's performance in images with relatively large people sizes and sparse crowd densities [34, 113, 46]. However, in complex scenes with serious occlusions and extremely crowded scenes, detection-based approaches often fail to detect individuals and hence produce inaccurate counting [137]. The feature-regression-based approaches, e.g., [107, 123, 67, 172], on the other hand, aim to obtain the density function of an image containing people and then calculate the total count by integrating the densities over the whole image space. They have demonstrated a countable solution for handling highly crowded scenes.

Recently, a Counting Convolutional Neural Network (CCNN) model [107] has been proposed, which can learn to count people and produce density maps in images. Compared with the traditional hand-crafted feature based approaches, this approach has achieved much better accuracy in wider, real-world crowded scenes. However, high capacity deep networks typically have significant inference costs especially when being used in complex scenes. This has resulted in a challenge for embedded sensors or mobile devices, where computational and power resources are often very limited. Many research works have been reported to reduce the storage and computation costs of deep neural networks for various applications. A typical solution is to prune the weights with small magnitudes and then retrain the network aiming not to downgrade the overall accuracy significantly [16, 101, 73, 43]. Yet, to our best of knowledge, no one has attempted to simplify the deep network models in a way that also improves their accuracy.

In this chapter, aiming to learn a lighter and more accurate deep network model, we propose a new strategy to prune the CCNN network [107] to not only simplify the network but also improve its accuracy. We examine the contributions of various filters in CCNN to the classification, and group the filters into positive, negative and irrelevant filters, respectively. Based on the feature maps of filters, we prune the irrelevant and negative filters so as to make the model lighter. Different from the

existing pruning algorithms, our goal is to not only reduce the size of the model, but also improve the performance through our proposed pruning strategy. When tested on benchmark datasets, our solution not only prunes the network but also improves the accuracy by removing non-contributing and negatively contributing filters.

The main contributions of our work are summarized as follows.

- We propose a new pruning strategy that not only prunes the network but also improves the accuracy without fine tuning.
- We propose a simple but effective mechanism to prune the irrelevant filters based on the feature maps which have little information, as well as the negative filters learned from training data.

The rest of the chapter is organized as follows. Section 4.2 shows the related work. In Section 4.3, the details about our proposed network pruning technique are given. The experiments conducted on various datasets are presented in Section 4.4. Finally, the chapter concludes in Section 4.5.

## 4.2 Related Works

Since detection-based counting approaches cannot be adapted to highly congested scenes, researchers try to deploy regression-based approaches to learn the relations between cropped image patches and their densities, and then calculate the number of particular objects. In recent years, many researchers [69] have developed deep learning models for image segmentation, classification and recognition, and have achieved very good results. Inspired by these, Convolutional Neural Network (CNN) models have been proposed to learn to count people and produce density maps in images simultaneously. These models work well for objects of a similar size in an image or a video. Sindagi and Patel [137] proposed an end-to-end cascaded network of CNNs that can learn globally relevant and discriminative features to

estimate highly refined density maps with low counting errors. Onoro-Rubio and Lopez-Sastre in [107] proposed a regression model called Counting CNN (CCNN) and the Hydra CNN for multi-scaled crowd counting. The CCNN and Hydra CNN can map the appearance features of input image patches to corresponding density maps.

Inspired by the Hydra CNN model, some researchers have tried to utilise more complex deep models to solve the problem caused by the significant variance of people's appearance in a captured image/video. Deepak et al. [123] proposed a switching CNN to select the best CNN regressor for each of the different receptive fields and achieved better results. Kumagai1 et al. [67] proposed a mixture of CCNNs and adaptively selected multiple CNNs according to the appearance of a testing image for predicting the number of people. Zhang et al. [172] proposed a multi-column network from three independent CNNs, and then used the combined features of these three networks to get a density map. Li et al. [76] proposed CSRNet by combining VGG-16 and dilated convolution layers to aggregate multi-scale contextual information. All of these works have suggested some effective solutions for counting people in complex real-world senses. However, all of these models require very high computation resources for running, creating a challenge for embedded or mobile systems to adopt these models. Therefore, it makes sense to reduce the network complexity.

Network pruning and sharing have been adopted to reduce the network complexity and address the over-fitting issue. A recent trend in this direction is to prune redundant or non-informative weights in a pre-trained CNN model. For example, Srinivas and Babu [142] explored the redundancies among neurons, and proposed a data-free pruning method to remove redundant neurons. Pavlo Molchanov et al. [101] proposed a new method to prune filters in neural networks. Li et al. [73] proposed to prune the filters that have little effect on the accuracy. The deep

compression method in [43] removed the redundant connections and quantized the weights, and then used Huffman coding to encode the quantized weights. In [148], a simple regularization method based on soft weight-sharing was proposed, and it included both quantization and pruning in one simple procedure. It is worthy to note that the above pruning schemes typically produce connection pruning in CNNs. However, all of these solutions achieve the pruning goal at the cost of losing accuracy to some extent.

For many cases, the networks may not have to be so complicated, so their complexity can be reduced. Then, is there any way to prune networks without decreasing their accuracy but with improved accuracy? It has been widely known that some filters contain little information for the final classification. However, according to our observation, some filters actually have negative impacts on the final classification. Therefore, pruning these filters will not only simplify the network models but also improve the network performance. In this chapter, we propose a pruning approach and demonstrate its superiority on the application of crowd counting.

## 4.3 Network Pruning

Our work presented in this chapter is initially designed for pruning the CCNN model [107] and can be applied to prune other crowd-counting network models. In this section, we only present the details of our proposed pruning strategy, as we introduced the CCNN-based crowd counting approach [107] in the Section 3.1.1 of Chapter 3.

### 4.3.1 Determining the Types of Filters

In training the CCNN model, the whole image is fed into the model. In crowd counting datasets, such as UCF and UCSD datasets, all of the images in training and testing datasets contain the target area, where the crowd is distributed, and

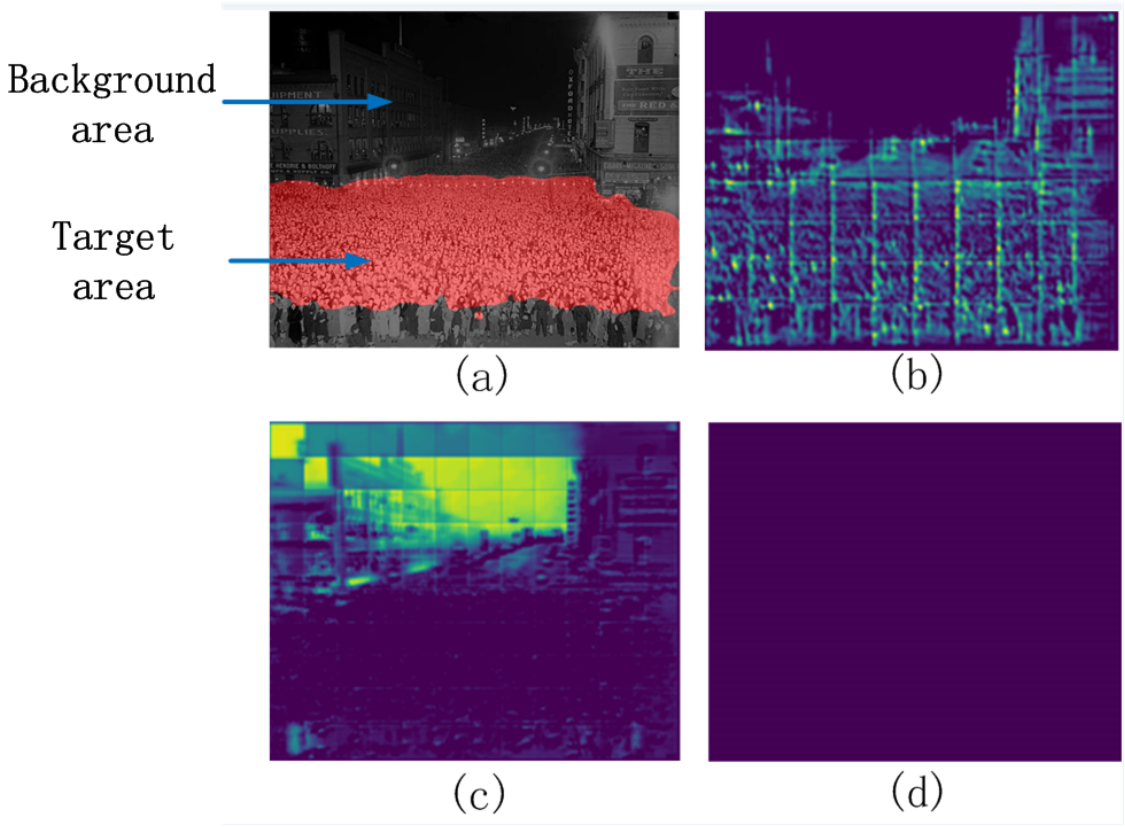



Figure 4.1 : **Example of the feature maps in layer conv4. (a) Input image. (b) Filters activating mostly on targets. (c) Filters activating mostly on background. (d) Filters with nearly no activation.**

the background area, where there are no people. According to [171], different filters activate on different targets of the images. Fig. 4.1 shows the activation of different filters in the feature maps corresponding to background and target areas, respectively.

In this figure, we can see that some feature maps have stronger activation on target area (see Fig. 4.1(b)), some filters activate mostly on background area (see Fig. 4.1(c)), and some feature maps contain nearly no activation (see Fig. 4.1(d)) and hence have little contribution to the classification result. Therefore, we can prune the model according to the activation of feature maps at different areas.

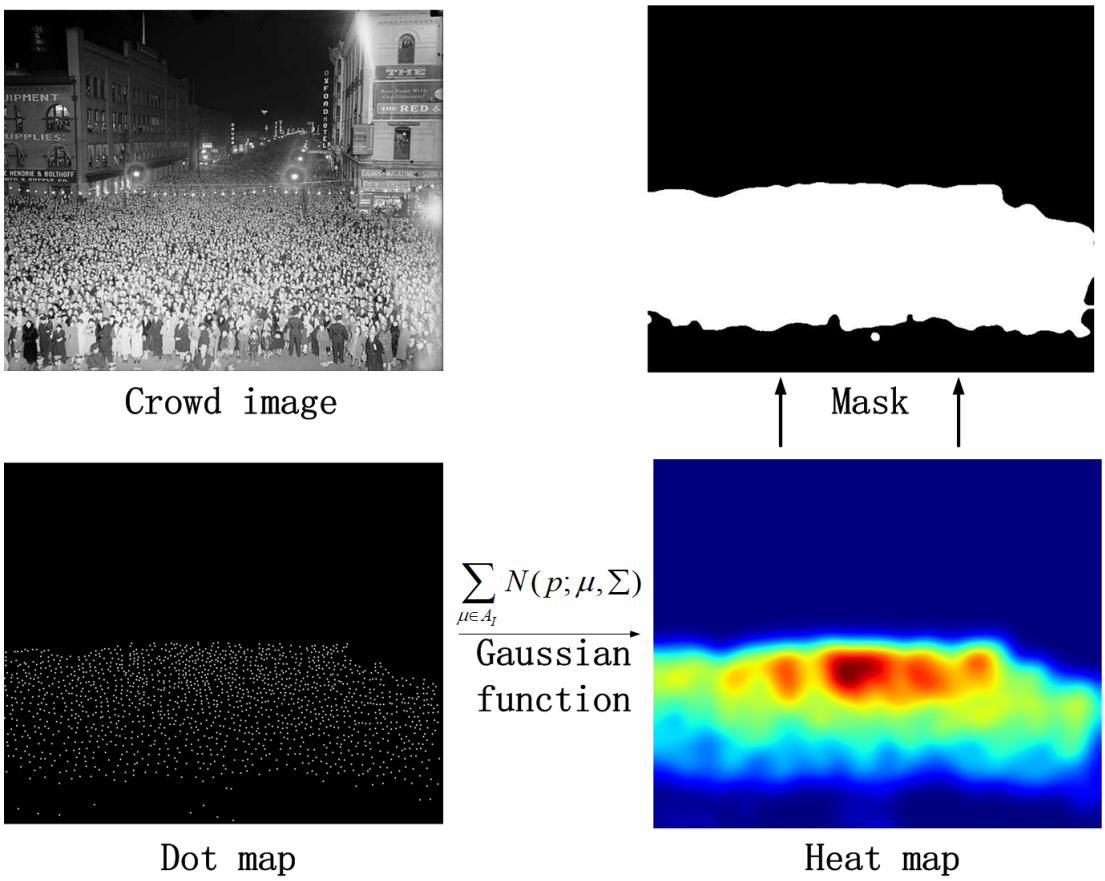



Figure 4.2 : **Learning the mask from an annotated training image.**

To examine the activations of feature maps corresponding to different areas, we learn a mask from annotated training images to identify the target area. Then, we define a simple mechanism to determine whether a filter makes positive or negative contributions to the classification, based on whether it mostly activates on target area or background area.

In a density map $D_I$, an intensity value larger than zero indicates that it has a non-zero density at the corresponding location. Thus, a binary mask, denoted by $M(x, y)$ (where $(x, y)$ is the coordinates of the pixel $p$), corresponding to a target (when its value is 1) and background pixel (when its value is 0), respectively, can

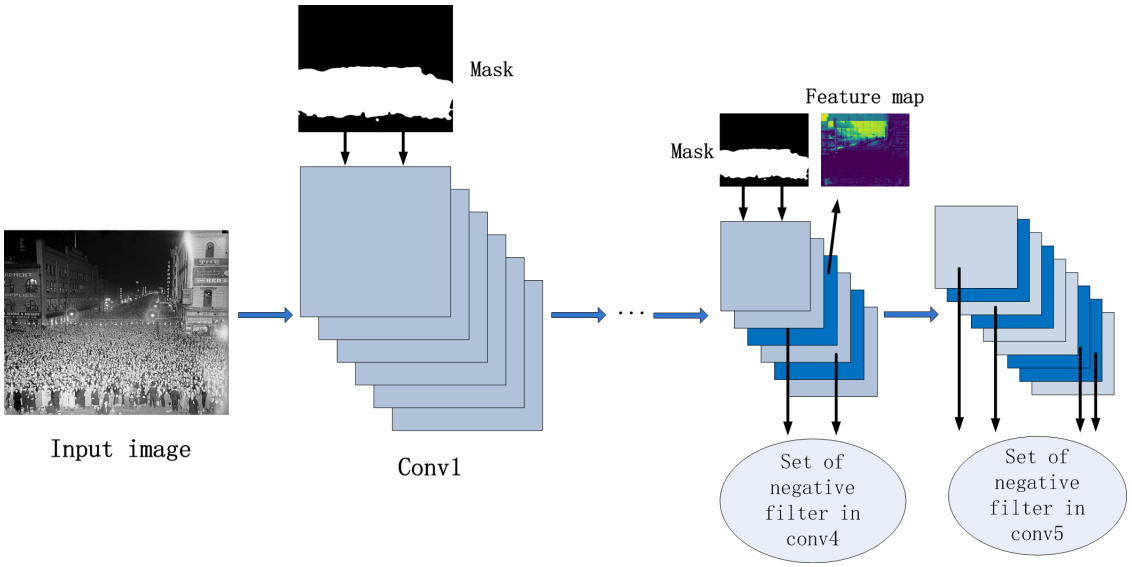



Figure 4.3 : **Model pruning with one mask.**

be derived from the density map function $D_I$ as:

$$M(x,y) = \begin{cases} 1, & \text{if } D_I(p) > 0; \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

Fig. 4.2 shows an example of the areas derived from the mask. In Fig. 4.2, the white area corresponds to the crowd area, and the black area represents the background. With the target and background learned using the mask, we can then easily determine whether a filter makes positive or negative contributions .

As shown in Fig. 4.3, after images are fed into the model, we apply the mask to all the feature maps in each layer. If the average magnitude of the background area (with the mask values equal to 0) divided by the average magnitude of the target area (with the mask values equal to 1) is higher than a pre-defined threshold $\eta$ (selected based on experiments), it is concluded that the corresponding filter activates more on the background than the target area (see Fig. 4.1(c)) and it is defined as a possible negative filter.

### 4.3.2 Pruning Filters and Feature Maps

Each training image has its own mask identifying its foreground and background, so it determines its own set of possible negative filters. In order to select the negative filters that are applicable to the entire dataset, we propose a simple voting mechanism to determine a maximum set of negative filters for the whole set of data. If a possible negative filter is included in most possible negative filter sets of training data, this filter will mostly likely be a negative filter for all data. Therefore, in this chapter, a filter is pruned if it is included in more than half of the negative filter sets.

To better illustrate this process, we take images from the UCF CC 50 dataset [55] . The UCF CC 50 dataset [55] consists of 50 pictures, collected from publicly available web images. Images in the UCF dataset [55] are randomly split into five subsets and a 5-fold cross-validation is performed by following the standard setting in [4].

We randomly take a 10-image set from the training set and then create their masks from their dotted annotation maps. Then, the resultant mask is applied to the corresponding training images. If filters activate on more than half (*i.e.*, 5 in this example) of the training images, the filter is determined to be a possible negative filter according to Sect. 4.3.1 and will be pruned. If the feature map contains nearly no information, this filter is determined to be an irrelevant filter and will also be pruned.

Table 4.1 shows the MAE results obtained on all sub-datasets in the UCF dataset [55] obtained using the CCNN models with and without pruning. As can be seen from this table that, after pruning, the accuracies are improved with the MAE reduced from 488 to 445. More comprehensive experiments are presented in Sect. 4.4.

Table 4.1 : **The MAE results obtained on all sub-datasets in the UCF dataset [55] obtained using the CCNN models with and without pruning.**

| | data0 | data1 | data2 | data3 | data4 | MAE |
|---|---|---|---|---|---|---|
| CCNN [107] | 775 | 476 | 510 | 276 | 373 | 488 |
| Pruned CCNN | **759** | **396** | **488** | **247** | **335** | **445** |

### 4.3.3 Pruning of Different Layers

For the CNN model, in the shallow layers, the filters extract basic features, such as edges, anchors and so on. While in the deep layers, the filters tend to extract high level features, such as those to identify heads and bodies [171]. Therefore, we do not prune the shallow layers of the model, and only prune deep layers. What is more, we also prune those filters without activations shown on the feature maps at all layers.

Fig. 4.4 shows the MAE results obtained for all five subsets of the UCF dataset [55] when all of he single layers are pruned without re-tuning. In this figure, the red line in each graph is the MAE of the original CNN model for each sub-dataset. As shown in this figure, pruning deep layers, *e.g.*, pruning Conv5 layer (shown as the purple bars in the chart) vs pruning Conv2 layer (shown as the blue bars in the chart), tends to have more impact to the performance of the overall model. On the contrary, pruning shallow layers (*e.g.*, Conv2 or Conv3) always has little effects on the performance. We can make a conclusion that, by pruning filters on deep layers, the counting results always get better. Thus, in our strategy, pruning is mostly carried out for Conv5 layer. Irrelevant filters containing little information in feature maps are pruned in Conv2 and Conv3, while in deep layers (Conv4 and Conv5), both irrelevant and negative filter are pruned.

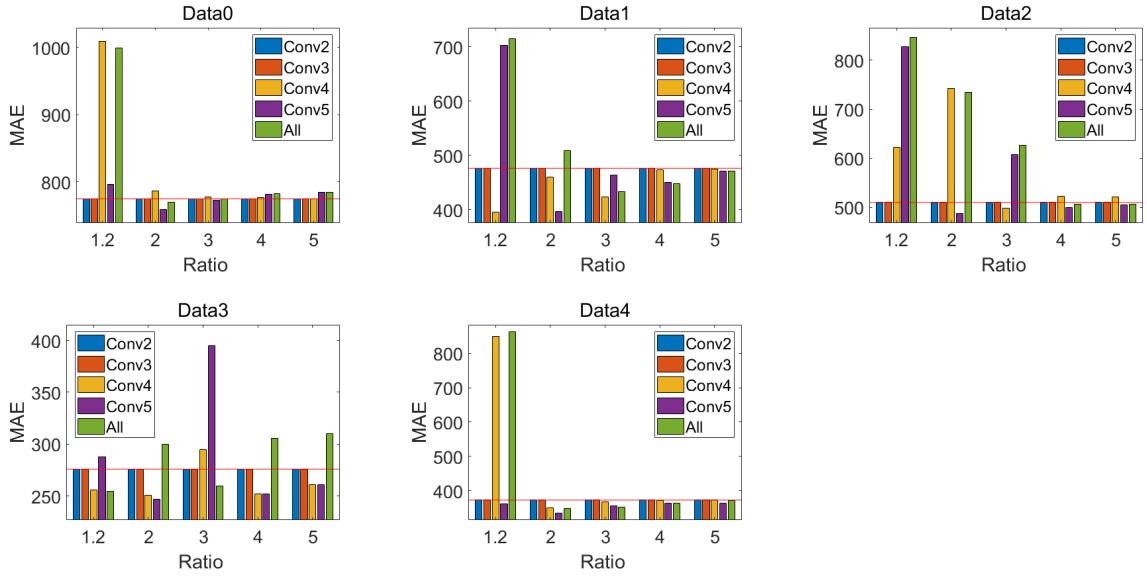



Figure 4.4 : **The MAE results of the estimations obtained on different subsets of the UCF dataset [55] by pruning different layers with different ratios $\eta$.**

## 4.4 Experiments

In this chapter, we evaluate and compare our proposed pruning mechanism on crowd counting CCNN networks on four widely used benchmark datasets, *i.e.*, the UCF [55], UCSD [11], Shanghai Tech [183] datasets, and the TRANCOS dataset [41]. We implement our filter pruning algorithm based on the Caffe deep learning framework. When filters are pruned, a new model with fewer filters is created and the remaining parameters of the modified layers as well as the unaffected layers are copied into the new model.

### 4.4.1 Comparison with the Original CCNN Model

Fig. 4.5 shows two estimated density heat maps and counts obtained with the original CCNN and our pruned CCNN on two exemplar crowd images. As it can be seen, the estimation obtained with our pruned CCNN is much more accurate.

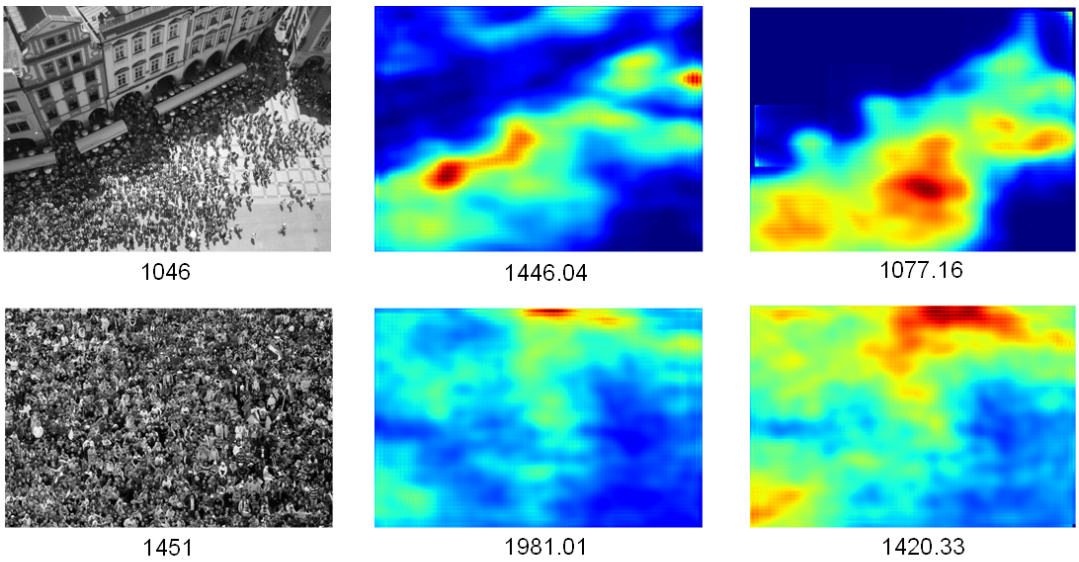



Figure 4.5 : **Examples of the density heat maps obtained with the original CCNN approach and our pruned CCNN model, where ground truth counts and estimation counts are shown underneath the images. (a) Input crowd images. (b) Density heat map obtained with the original CCNN. (c) Density heat map obtained with our pruned CCNN.**

Next, following the convention of the similar works [123, 126, 67, 172] for crowd counting, we evaluate the performance of different approaches quantitatively on two datasets using the MAE, which is defined by the equation 2.1. Roughly speaking, the lower the MAE is, the better accuracy the estimation method has.

***Experimental Results on the UCSD Dataset***

The UCSD dataset [11] provides the Region of Interest (ROIs) for each video frame. We use the ROI as the mask to determine the type of filters. As the scene of UCSD is fixed, we use one mask and follow the rules in the UCF to prune the model. The results are shown in Table 4.2. Note that the size of the original model is 2.3MB. As shown in this table, after the pruning, the sizes of the models for the four sub-

Table 4.2 : **Comparison of the MAE results on the UCSD [11] dataset.**

| | maximal | downscale | upscale | minimal |
|---|---|---|---|---|
| CCNN [107] | 1.70 | 1.79 | 1.13 | 1.50 |
| **Our pruned CCNN** | **1.63** | **1.70** | **0.96** | **1.49** |
| **Pruned Model size/MB** | 1.5 | 1.3 | 1.3 | 1.5 |

Table 4.3 : **Comparison of the MAE results TRANCOS [41] dataset.**

| Method | GAME0 | GAME1 | GAME2 | GAME3 |
|---|---|---|---|---|
| CCNN | 12.49 | 16.58 | 20.02 | 22.41 |
| Pruned CCNN | **11.25** | **14.26** | **16.43** | **19.72** |
| **Pruned Model size/MB** | 1.7 | 1.1 | 2.1 | 1.8 |

datasets are $1.5MB$, $1.3MB$, $1.3MB$ and $1.5MB$, respectively, decreased by 35% to 44%. Moreover, the accuracy obtained on all four sub-datasets are improved to some extents with reduced MAEs.

***Experimental Results on the TRANCOS Dataset***

Table 4.3 reports the MAE results obtained on TRANCOS [41] dataset with the original CCNN model and our pruned model. As it can be seen from this table, the crowd count using the pruned CCNN is significantly higher than that of the original CCNN.

### 4.4.2 Comparison with Other Pruning Algorithms

We compare our proposed algorithm with other pruning algorithms applied to the CCNN model, *i.e.*, [73] and [94]. As the UCSD and UCF are the only two

Table 4.4 : **Comparison of the proposed algorithm and other pruning algorithms on CCNN.**

| Model/DATA | CCNN | [73] | ThiNet [94] | Distillation [48] | Our algorithm |
|---|---|---|---|---|---|
| UCSD maximal | 1.70 | 1.73 | 1.72 | 1.72 | **1.63** |
| UCSD minimal | 1.50 | 1.50 | 1.51 | 1.55 | **1.49** |
| UCSD upscale | 1.13 | 1.14 | 1.14 | 1.11 | **0.96** |
| UCSD downscale | 1.79 | 1.78 | 1.81 | 1.84 | **1.70** |
| UCF data0 | 775 | 775 | 782 | 768 | **759** |
| UCF data1 | 476 | 476 | 450 | 483 | **396** |
| UCF data2 | 510 | 510 | 515 | 529 | **488** |
| UCF data3 | 276 | 276 | 279 | 293 | **247** |
| UCF data4 | 373 | 373 | 377 | 364 | **335** |

datasets for crowd counting based on CCNNs, we demonstrate the comparison on these two datasets.

Table 4.4 reports the MAE performance obtained using the proposed pruning algorithm and the algorithms proposed in [73] and [94]. As shown in this table, the MAE of our pruned CCNN on both datasets UCSD and UCF are significantly better than those of the original CCNN and the pruned CCNN with other pruning algorithms. Note that the results of [73], ThiNet [94] and Knowledge Distillation [48] are similar to those of the original CCNN and hence almost do not show any significant improvement on accuracy. However, our proposed pruning algorithm can not only reduce the size of the model, but also improve the results of the original CCNN model.

Table 4.5 : **Comparison of the results obtained on the Shanghai Tech dataset using the MCNN and Switch-CNN counting models with and without applying our pruning method.**

| | partA | | partB | |
|---|---|---|---|---|
| Method | MAE | MSE | MAE | MSE |
| MCNN [183] | 110.2 | 173.2 | 26.4 | 41.3 |
| **Pruned MCNN** | **100.5** | **170.5** | **23.5** | **39.7** |
| Switch-CNN[123] | 90.4 | 135.0 | 21.6 | 30.1 |
| **Pruned Switch-CNN** | **89.5** | **136.2** | **21.5** | **32.3** |
| Pruned MCNN size/KB | 425 | | 462 | |
| Pruned Switch-CNN size/KB | 436 | | 477 | |

### 4.4.3 Pruning Results on Other Crowd Counting Models

To demonstrate that our pruning method can also work with other models, we use the same method to prune other crowd counting models, *i.e.*, the MCNN model [183] and Switch-CNN models [123], on the Shanghai Tech dataset [183], UCF dataset [55] and UCSD dataset, respectively. Note that, different from other pruning algorithms, we do not fine-tune or re-train our new model after pruning, but it still produces better results. The results are shown in Tables 4.5 and 4.6. Note that, in order to compare with MCNN and Switch-CNN, we add another metric, *i.e.*, MSE that we defined by the equation 2.2.

As shown in these two tables, both of the MAE and MSE results obtained with the pruned MCNN and pruned Switch-CNN are significantly better than with the original MCNN and Switch-CNN, respectively. Moreover, the sizes of the original MCNN and Switch-CNN are 515KB, while the size of pruned models are decreased

Table 4.6 : **Comparison of the results obtained on the UCF and UCSD datasets using the MCNN and Switch-CNN counting models with and without applying our pruning method.**

| | UCF | | UCSD | |
|---|---|---|---|---|
| Method | MAE | MSE | MAE | MSE |
| MCNN [183] | 377.6 | 509.1 | 1.07 | 1.35 |
| **Pruned MCNN** | **326.5** | **472.3** | **1.02** | **1.31** |
| Switch-CNN[123] | 318.1 | 439.2 | 1.62 | 2.10 |
| **Pruned Switch-CNN** | **305.1** | **410.9** | **1.44** | **1.73** |
| Pruned MCNN size/KB | 413 | | 389 | |
| Pruned Switch-CNN size/KB | 503 | | 495 | |

at different degree. This demonstrates that our algorithm can not only work on CCNN, but also on other counting models.

### 4.4.4 Impact of $\eta$

Moreover, we use a ratio learned from the dataset to determine the contribution of the filters in order to optimize the pruning effectiveness. In our work, the $\eta$ is determined statistically through experiments. Fig. 4.4 shows the MAE results obtained on each of the five subset of the UCF dataset [55] with different ratio $\eta$.

According to Fig. 4.4, the performance of different sub-models on this dataset achieves the best when using a ratio 2 to prune the Conv5 layer.

## Acknowledgment

This work was contributed equally by Lei Liu (a visiting PhD student of Professor Sean He at UTS). Lei and the author of this thesis have contributed equally to the

design and implementation of the research idea, to the analysis of the experimental results and to the writing of the manuscript. This chapter has been published at the Neurocomputing journal (with title 'Performance-enhancing network pruning for crowd counting' [82]).

## 4.5 Conclusion

In this chapter, we have proposed a new pruning strategy for crowd counting that works with CCNN and other crowd counting models. Through identifying positive, negative and irrelevant filters according to the activation of feature maps, our solution has not only reduced the network size but also improved the accuracy by removing non-contributing and negatively contributing filters. Experimental results on benchmark datasets have shown that, compared with other existing pruning algorithms, our proposed technique can improve the accuracy of counting models without fine-tuning or retraining the pruned model, and meanwhile reduce the size of the models.

# Chapter 5

# PDANet: Pyramid Density-aware Attention Net for Accurate Crowd Counting

In this chapter, we propose a novel Pyramid Density-Aware Attention-based network, abbreviated as PDANet, which leverages the attention, pyramid scale feature, and two branch decoder modules for density-aware crowd counting. The PDANet utilises these modules to extract different scale features, focus on the relevant information, and suppress the misleading ones. We also address the variation of crowdedness levels among different images with an exclusive Density-Aware Decoder (DAD). For this purpose, a classifier evaluates the density level of the input features and then passes them to the corresponding high and low crowded DAD modules. Finally, we generate an overall density map by considering the summation of low and high crowded density maps as spatial attention. Meanwhile, we employ several losses to create a precise density map for the input scene. Extensive evaluations conducted on the challenging benchmark datasets well demonstrate the superior performance of the proposed PDANet in terms of the accuracy of counting and generated density maps over the well-known state-of-the-art approaches.

## 5.1 Introduction

Nowadays, crowd counting has become an important task for a variety of applications, such as traffic control [88], public safety [79], and scene understanding [127, 79]. As a result, density estimation techniques have become a research trend for various counting tasks. These techniques utilise trained regressors to estimate people density for each area so that the summation of the resultant density func-

tions can yield the final count of crowd. A variety of regressors, such as Gaussian Processes [12], Random Forests [71], and more recently, deep learning based networks [83, 72, 82] have been used for crowd counting and density estimation. However, the state-of-the-art approaches are mostly deep learning based approaches due to their capabilities of generating accurate density maps and producing precise crowd counting [88, 178].

Generally, the approaches based on deep neural networks (DNNs) utilise standard convolutions and dilated convolutions at the heart of the models to learn local patterns and density maps [82, 2]. Most of them use the same filters, pooling matrices, and settings across the whole image, and implicitly assume the same congestion level everywhere [83]. However, this assumption often does not hold in reality.

To better understand the effect of this mis-assumption, let us show some examples with clearly different levels of crowdedness. Fig. 5.1 presents some exemplar images of different congestion scenarios. Fig. 5.1(a) shows a highly crowded image having more than 1,000 people, while Fig. 5.1(c) presents a less crowded scene having less than 70 people. However, if we look at Fig. 5.1(a), we notice that there is a relatively more congested area, which is shown in Fig. 5.1(b). The same situation can be seen in Fig. 5.1(c), and it is obvious that a small area within this crowd, as shown in Fig. 5.1(d), is more crowed. Due to this dynamic variation in the crowded scenes, naturally we should utilise different features and branches to respond and capture details at different levels of crowdedness. In the past, this has been attempted by four major types of approaches, *i.e.*, defining separate pathways from the lower layers and utilizing different sizes of the convolutional filters, image pyramid-based methods [15, 88], detection-based crowd counting [83], patch-based crowd counting [123, 2, 122], and multi-level feature based methods [15]. Although these methods achieved robust performance with some different tactics, there are still lots of spaces to improve their performances by designing highly efficient convo-

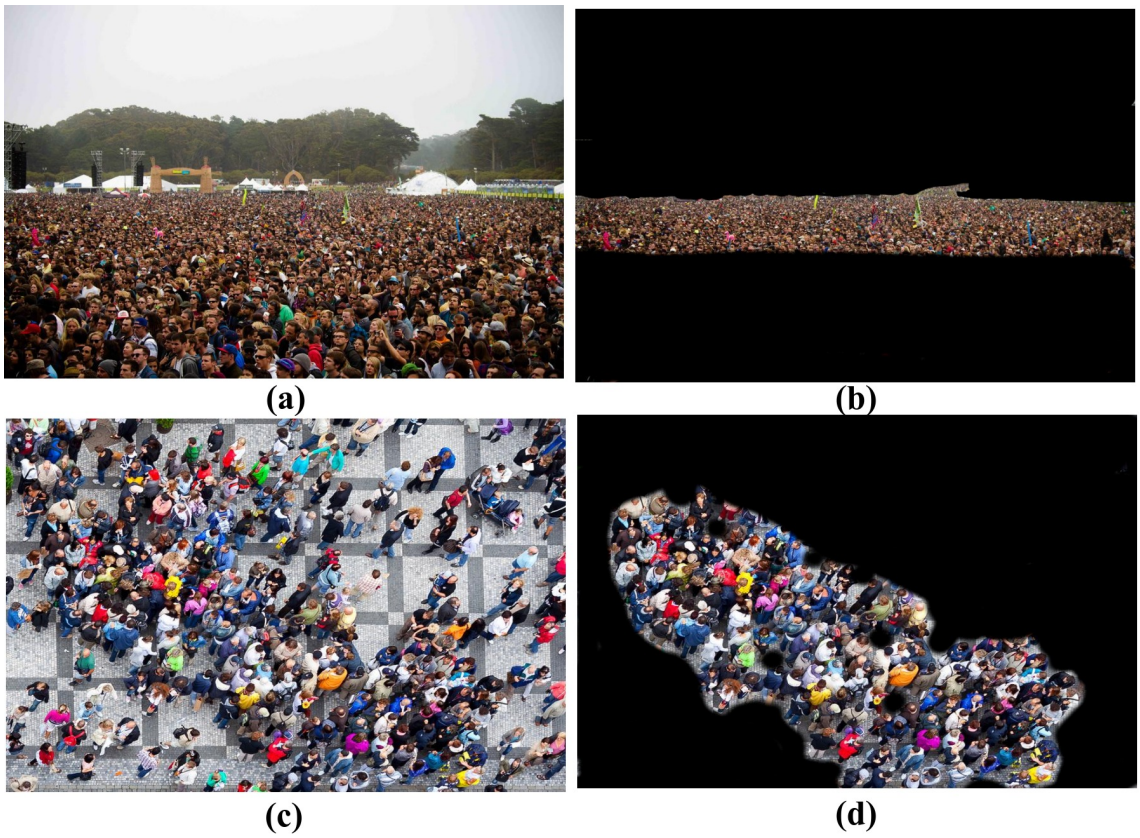



Figure 5.1 : **Examples of crowded and sparse images. (a) and (c) show an example of a highly crowded scene and a less crowded scene, respectively, while (b) and (d) show their corresponding congested areas.**

lutional layer structures, which can effectively deal with crowd scenes with dramatic density varieties effectively.

First, generally speaking, a kernel size of $3 \times 3$ for a convolution filter is more effective than the larger ones in terms of extracting more meaningful features, because more details can be captured with lower complexities without making it more difficult to train the network [134, 143, 144]. Kang *et al.* [63] proved that smaller receptive fields gave better performance. Secondly, using patch-based processing and multi-patch processing is time costly due to that the same features have to pass through different paths and patches multiple times. If we want to take the benefits

of multi-patch or multi-branch based approaches, it is better to extract some coarse features from the initial layers and then pass them to some branches for further zooming in to find more sophisticated features. To utilise a deeper network for crowd counting, we need an approach that can deploy the aforementioned proposals on the multi-branch structure to achieve better performance.

In this chapter, we present a deep encoder-decoder based architecture named as Pyramid Density-aware Attention-based Network (PDANet), which combines the pyramid feature extraction with spatial and channel attentions to produce richer features estimating crowd of various levels of crowdedness and scales. In our work, we use the VGG16 as the feature extractor for the encoder to produce features for the decoder of the model. To learn multi-scale features, we first use a cascade of Global Average Pooling (GAP), $1 \times 1$ convolution and dilated convolutions with kernels of $3 \times 3$ to extract more mature features with different scales from VGG16 features. Then, we apply the channel and spatial attentions in different layers to enhance and boost the quality of features in order to obtain more accurate density maps. On the other hand, to make the model adaptive to different density levels within an image, we introduce a classification module to classify the crowdedness level of the input scene and develop generation models of low and high crowded density maps for the input image.

This work is different from the existing crowd counting approaches that use the pyramid contextual information and attention modules in several ways. (a) Unlike our previous model, DENet [83], the proposed PDANet does not separate models for counting people in sparsely crowded areas and estimating the human density maps in the remaining areas in an image. (b) The first main characteristic of our proposed PDANet is its density awareness by adopting the pyramid and attention modules. Different from other works attempting to address this problem of density variety, *e.g.* [123], our PDANet does not separate the input scene into

different patches. Instead, we use multipath branching to address the intra-density variations within the input scene. Experimental results show that the pyramid and attention modules contribute a 5 to 20 percent improvement over the baseline model. (c) Pyramid Feature Extractor (PFE) is the second noticeable contribution of our PDANet. We utilise a new combination of GAP, $1 \times 1$ convolution, and Atrous convolution, resulting in a difference from the existing approaches in terms of the orders and parameters that can better aggregate local scale features and is more effective than the existing solutions. (d) The third remarkable feature of PDANet is its attention modules. The architecture of our end-to-end attention modules is also different from ADCrowdNet [86] because it uses the combination of the spatial and channel-based attention modules within the architecture. Furthermore, it is trained in an end-to-end way based on the crowd counting dataset, instead of separately using the external dataset to train the attention module as in ADCrowdNet [86]. Compared with the work in [162], our PDANet has also adopted another spatial-based module in the DAD module to optimize the density map results based on feature maps of the sparse and dense areas within the input scene. (e) The last distinct characteristic of the PDANet is classification modules, which are different from the existing work [162]. Our PDANet passes the input image to two different sub-models with different receptive fields to evaluate lower and higher bounds of the density map and then combines them with the help of the channel attention module. Our PDANet introduces a classification module that classifies the input image to the low or high-density data and passes them to the corresponding and appropriate DAD modules.

To summarize, the contributions made in this chapter are as follows.

- In order to address crowd areas of various scales and density levels, we propose a density-aware solution, which is achieved with the combination of multi-scale feature extraction, density classification and adaptive density estimation

modules. This feature helps the model to handle density variation between different images as well as within each input scene.

- We first integrate the pyramid multi-scale feature extraction mechanism in a feature extractor to extract rich features for the following classification module. Then, we integrate the channel and spatial attention modules and propose an end-to-end trainable density estimation pipeline. Both modules contribute to exploit the right context at each location within a scene.

- For estimating densities of crowd with not only high and low crowdedness levels but also inter-level density areas, we propose to use a combination of classification and regression losses to address the whole and within-the-scene changes in the density maps.

- Extensive experiments on several challenging benchmark datasets are conducted to demonstrate the superior performance of our proposed PDANet approach over the state-of-the-art solutions. We also preform comprehensive ablation studies to validate the effectiveness of each component in our proposed approach.

The rest of the chapter is organized as follows. In Section 5.2, we introduce the existing works related to our approach. The proposed PDANet model for crowd density estimation is introduced in detail in Section 5.3. In Section 5.4, we evaluate the performance of PDANet on benchmark datasets. Section 5.5 provides ablation studies on various parts of the proposed model. Finally, we draw conclusions in Section 5.6.

## 5.2 Related Works

In this section, we provide literature review related to our PDANet model. As we discussed in the section 2.2.4, there are two common network architectures for crowd

counting, multi-branch [183, 123, 136, 24, 146] and single-branch network [143, 184, 76, 9, 129]. As we explained in the section 2.2.4, mutli-branch methods, attention modules can help us to reach a robust and accurate density estimation model.

Most recently, Shi et al. [130] proposed a perspective information CNN-based model PACNN for crowd counting. Their model combined the perspective information with a density regression to address the person scale change within an image. They generated the ground truth of perspective map and used it for generating perspective-aware weighting layers to combine the results of multi-scale density adaptively. Wan et al. [153] proposed a new model RRSP to utilise the correlation information in a training dataset (residual information) for accurate crowd counting. They fused all the residual predictions and created the final density map based on the appearance-based map and the combination of residual maps from the input scene.

Although the idea of using the attention map in ADCrowdNet [86] was interesting, it has some significant drawbacks, such as that (a) it needed an external dataset to train AMG to detect the crowd area, and (b) after producing the attention map, they applied it on the input scene to create masked input data for DME and again extract features with a similar encoder-decoder structure. We believe that it is redundant and time consuming due to passing the input scene twice rather than applying the generated mask on the latest layer of AMG module and using the feature maps for the next stage.

## 5.3 Pyramid Density-aware Attention Net

In this section, we first present the general structure of our proposed PDANet for adaptively addressing the challenges in crowd counting. This new structure uses pyramid-scale feature extraction and consists of adaptive pooling, and $1\times1$ and $3\times3$ convolutions to enrich the feature maps for handling objects of various scales within

a scene. In the following subsections, we will give more details about the Attention module, Pyramid Feature Extractor module, Classification module, Density Aware Decoder module and loss functions.

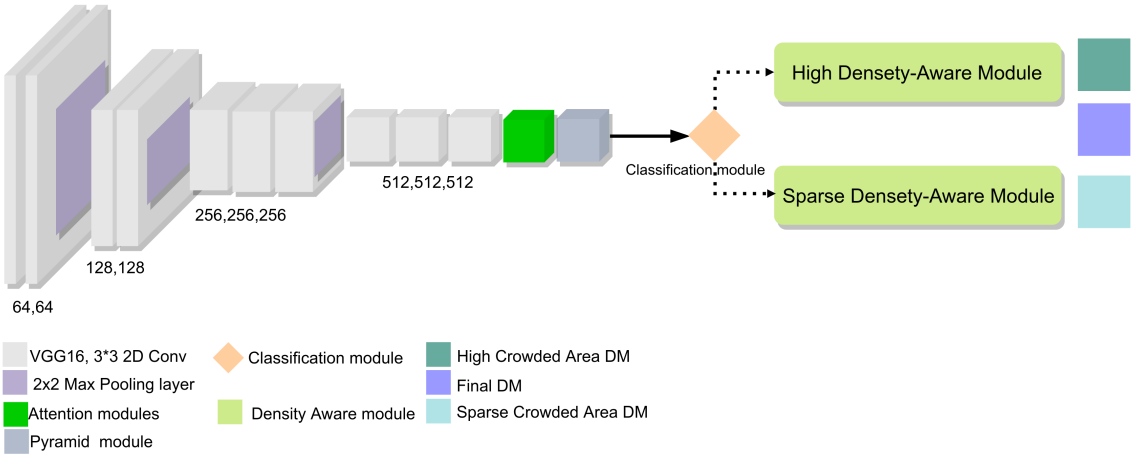



Figure 5.2 : **The overview of our proposed PDANet network. This architecture contains a VGG16 based feature extractor, a Pyramid module, an Attention module, a Classification module, and a Decoder module.**

### 5.3.1 Overview

As discussed above, we formulate crowd counting as the problem of regressing people's density map from a scene. The overall architecture of our PDANet for regressing the density map of the crowd from an image is illustrated in Fig. 5.2. This framework contains five main components, *i.e.*, a Feature Extractor, a Pyramid Feature Extractor (PFE), a Classifier, a Density Aware Decoder (DAD), and an Attention Module. Each of these components contributes to the overall accuracy and efficiency of the model for crowd counting.

The backbone of our PDANet is a network based on VGG16 [134], which is widely used for extracting low-level features. We eliminate the layers between the last two pooling layers considering the trade-off between resource cost and accuracy [76].

Then, we apply a channel and spatial based attention module to it to highlight essential features. Then, these features are fed into the PFE module, which incorporates the combination of adaptive pooling and $1 \times 1$ and $3 \times 3$ dilated convolutions to produce scale-aware mature features for last layers of the decoder module. In the next step, we incorporate a GAP and a fully connected layer to classify the input scene as a highly dense or a sparse one. Then, we pass this information to the respective decoder with the same structure (our theoretical studies proves that the same respective field is better than a different one). The decoder contains four $3 \times 3$ dilated convolution layers, which are empowered with an attention module after each layer. Furthermore, to address the congestion differences in sparse and dense areas, we design two branches of the decoder module to generate low and high-density maps within the input scene and assign them to the corresponding regression losses. In the final step, we use the dense and sparse features from the last layer of the decoder to produce the final output density map (DM). Our PDANet uses the same loss for sparse, dense and final output DMs, and a classification loss to train the model in an end-to-end manner.

To summarize, in our proposed PDANet, each part plays a role in the overall performance.

- The Attention Module focus its attention on the significant features (crowded areas).
- The Pyramid Feature Extractor generates more productive features, which are more suitable for the crowd counting task with scale variation, through a combination of adaptive pooling algorithms and dilated convolutions with different scales.
- The Classifier helps find the proper branch of the decoder according to the crowdedness level of the area.

- The mid-branch Decoder is to address congestion change within the input image.

### 5.3.2 Channel and Spatial based Attention Modules

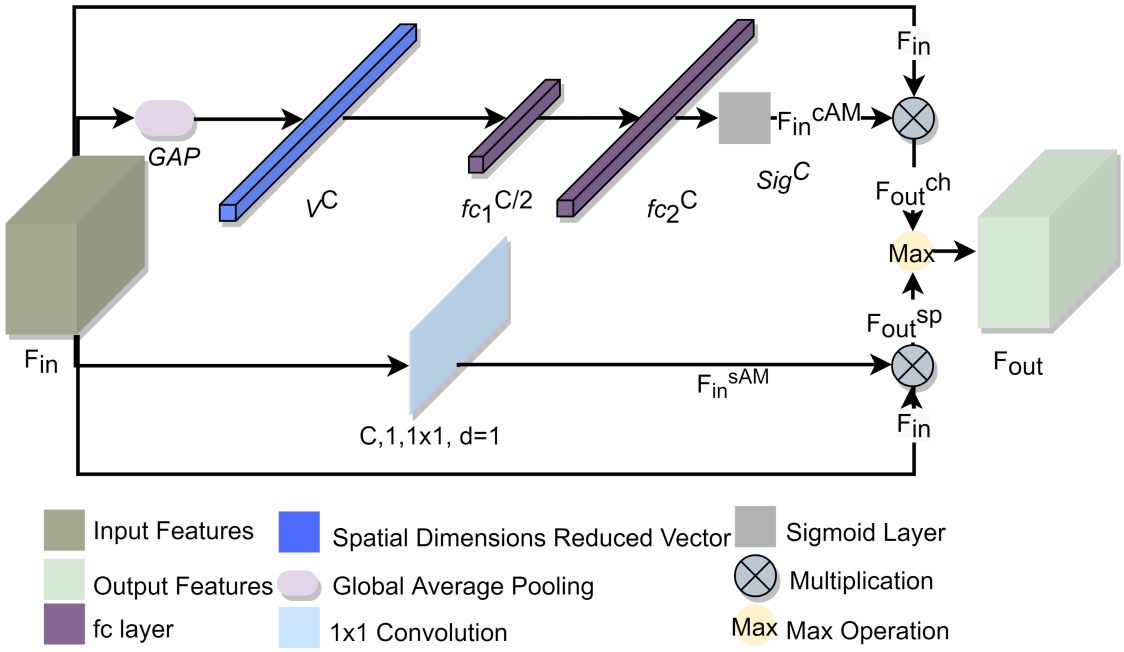



Figure 5.3 : **Illustration of the attention module of our model. The top branch generates channel-based attention, while the bottom branch generates the spatial attention map.**

In this study, we re-calibrate the feature maps adaptively by mixing attention modules to augment the effect of essential features, while suppressing the weak ones. We use the combination of spatial and channel-based attentions for finding and separating the crowded areas within the input image. As it is shown in Fig. 5.2, we utilise an attention module in our model, which is the channel and spatial attention [118] after the convolution layers, shown as the green module in Fig. 5.2. This module contains the channel and the spatial attentions to produce the final attention features in each layer. We combine the results of these two attentions by an

element-wise max of the channel and the spatial excitation to generate output features in each layer. The other attention module is a spatial attention map that is generated based on the density map of the sparse and dense crowded areas within the image. We apply a sigmoid on this attention module and multiply it with the joint convolution feature maps from the last layer of a sparse and dense decoder.

Fig. 5.3 illustrates this attention module. As shown in this figure, there are two branches in this illustration, *i.e.*, the channel attention branch on the top, and the spatial attention branch on the bottom. The channel attention branch utilises a cascade of GAP and two fully connected layers with the size of $\frac{C}{2}$ and $C$, respectively ($C$ is the channel size of a convolution layer). Then, after applying a sigmoid on the result, we do element-wise multiplication between the channel attention map and the input feature maps to obtain channel-wise weight corrected feature maps.

As we explained in the previously, to apply the channel based attention mechanism, we first perform GAP on the input feature map $F_{in}$, to obtain $V^C$, and then transform them by two fully-connected layers $f_{c1}^{C/2}$ and $f_{c2}^{C}$, as shown in Fig. 5.3 and Eq. 5.1 as:

$$F_{in}^{cAM} = \text{Sig}(f_{c2}^{C}(f_{c1}^{C/2}(V^C))). \tag{5.1}$$

where Sig is a sigmoid function that yields the value in a range of $[0, 1]$ to find the impact of each layer in the feature maps.

Therefore, for channel based attention, features $F_{out}^{ch}$ are obtained by multiplying the encoded channel-wise dependencies ( $F_{in}^{cAM}$ ) with $F_{in}$ to get $F_{out}^{ch}$. On the other hand, to obtain the spatial attention map, $F_{in}^{sAM}$, we perform a $1 \times 1$ convolution, *i.e.*, $Conv \in \mathbb{R}^{1\times C\times 1\times 1}$, on the input feature map. Thus, we can measure the importance of a spatial information of each pixel or location within $F_{in}$. In the next stage, we multiply the spatial attention map with the input feature maps to get the final spatial attention features $F_{out}^{sp}$, which augment relevant spatial locations and

suppress irrelevant ones. Finally, we combine the results of these two attentions by element-wise max of the channel and spatial excitation, *i.e.*, $F_{out} = \max(F^{ch}_{out}, F^{sp}_{out})$. These feature maps amplify the input feature map data and re-calibrate the crowded area within each input convolution layer.

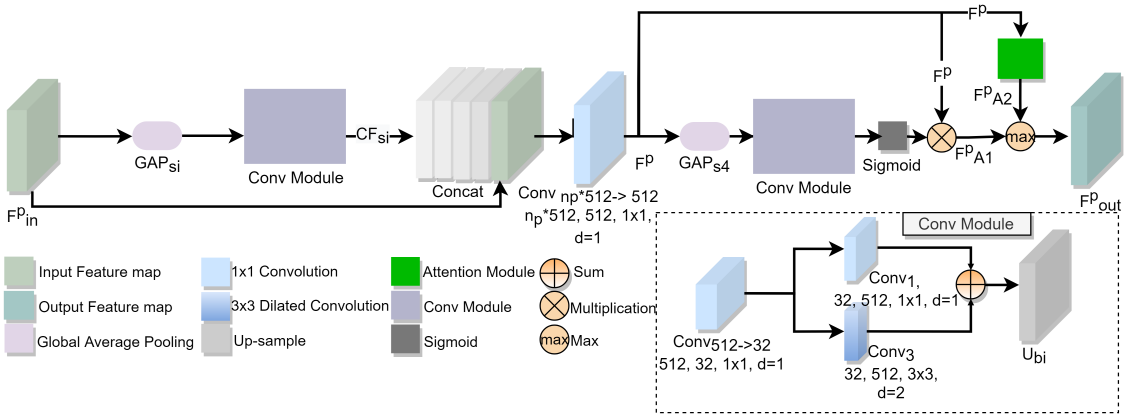



Figure 5.4 : **The overview of the Pyramid Feature Extractor (PFE) module. The PFE module uses $1 \times 1$ and $3 \times 3$ dilated kernel convolutions with the GAP to extract features of different scales from the VGG16 features.**

### 5.3.3 Pyramid Feature Extractor (PFE)

In this section, we propose a Pyramid Feature Extractor (PFE), which is inspired by the Spatial Pyramid Pooling [47] to address this issue. The PFE fuses features under various pyramid scales by a combination of GAP and two shared 2D convolution layers with a mixture of $1 \times 1$ and $3 \times 3$ dilated kernels. The general operation of PFE is illustrated in Fig. 5.4.

We extract contextual features by various GAP. In PFE module, we keep the ratio of the input feature map with $\text{GAP}_{s_i}$ at scale $s_i$, for $i = 2, 3, \cdots, 10$ and produce contextual features for each channel with a size of $Hs_i \times W_{s_i}$. For example, if we have an input feature map with $\mathbb{R}^{1 \times C \times H \times W}$, $\text{GAP}_{s_2}$ utilise global average pooling layer to generate scaled feature map with size of $\mathbb{R}^{1 \times C \times \frac{H}{2} \times \frac{W}{2}}$, where $H_{s_2}$ and $W_{s_2}$

are equal to $\frac{H}{2}$ and $\frac{W}{2}$, respectively. Various scales of contextual features form the pooled representations for different areas and provide rich information about the density levels in various sub-regions of the input image. The results presented in the Experiments section are based on the scenario utilizing three $\text{GAP}_{s_i}$ with scale of $s_2$ , $s_4$ , and $s_8$, respectively. In the Ablation Study shown in Section 5.5, we compare several scenarios for the use of $\text{GAP}_{s_i}$.

Then, we feed $\text{GAP}_{s_i}$ to the Conv Module to improve the representation power of the feature map. This procedure is different from the architectures that reduce the dimension of the input feature map with convolution [15]. As illustrated in Fig. 5.4, we perform the Conv operation as:

$$\text{CF}_{s_i} = U_{bi}(\text{Conv}_1(\text{Conv}_{512\to32}(\text{GAP}_{s_i})) + \text{Conv}_3(\text{Conv}_{512\to32}(\text{GAP}_{s_i}))). \quad (5.2)$$

where, for each scale $s_i$, $\text{CF}_{s_i}$, is the shared Conv module that comes with a bi-linear interpolation to up-sample the contextual features $U_{bi}$ to the ones of the same size as that of $F_{in}^p$.

The shared layer contains one $1 \times 1$ convolution ($\text{Conv}_{512\to32}$) to reduce the number of channels from 512 to 32. We do this to reduce the number of parameters that need to train and reduce the computational cost of PFE. In the subsequent stage, we get the summation of a $1 \times 1$ convolution ($\text{Conv}_1$), and a $3 \times 3$ dilated convolution ($\text{Conv}_3$) as a piece of extra bonus information that captured from surrounding contextual features within $\text{GAP}_{s_i}$. Experimentally, we verify that this combination of convolution filters improves the performance of the PFE module in the density estimation task. Finally, we concatenate all $\text{CF}_{s_i}$ and the input features $F_{in}^p$ with a $1 \times 1$ convolution. We reduce the number of the channels to the original VGG features $F_{in}^p$. We define this by:

$$F^p = \text{Conv}_{n_p*512\to512}(\text{Concat}(\text{CF}_{s_i}, F_{in}^p)), \quad (5.3)$$

where $n_p$ is the number of pyramid contextual features $\text{CF}_{s_i}$, plus the features in the

original feature map, and $\text{Conv}_{n_p*512\rightarrow512}$ is a 1 convolution to reduce the number of channel to 512.

Then, we utilise a special attention module, which is the combination of the Conv module and attention module that we explained in Section 5.3.2. We pass $F^p$ to two separate attention branches. As illustrated in Fig. 5.4, in the bottom, we feed $F^p$ to the $\text{GAP}_{s_4}$ layer and reduce the input size to $\frac{H}{4} \times \frac{W}{4}$, and then apply the Conv module to it. We apply the GAP to highlight and escalate the most important parts of the output feature maps. Then, after performing Sig on the output of conv module, we apply the element-wise multiplication to produce the attention map by Conv module ($F^p_{A1}$). On the top, we also perform the attention module that we discussed in the Subsection 5.3.2 to generate the attention feature map output ($F^p_{A2}$). Finally, we combine the results of these two attentions by the element-wise max operation of the Conv module output ($F^p_{A1}$) and the attention module output ($F^p_{A2}$) as defined by:

$$F^p_{out} = \max(F^p_{A1}, F^p_{A2}). \tag{5.4}$$

Altogether, as illustrated in Fig. 5.4, the PFE module extracts the contextual features $\text{CF}_{s_i}$ as discussed above, and then feed them to the classification module and a Density Aware Decoder (DAD) module that produces the density map.

### 5.3.4 Classification Module

The next step in our overall framework, as illustrated in Fig. 5.2, is to decide whether the input contextual features are dense or sparse. We do this to address the huge variation of crowd densities among different images. We pass input features to the suitable DAD to adaptively react to the density level of the input image and provide a better estimation for crowd density. To model this, as shown in Fig. 5.5, we introduce a binary classification module to learn how to classify the input feature

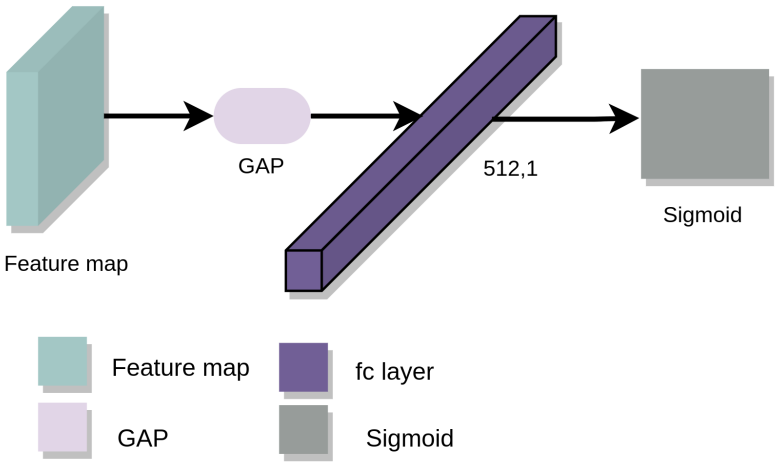



Figure 5.5 : **Illustration of the classification module of PDANet. It uses the global average pooling with a fully connected layer to determine the dense level of input scene.**

maps into two classes, *i.e.*, dense or non-dense (*aka*, sparse), as:

$$\mathrm{Cl}_t^{est} = \mathrm{Sig}(f_c(\mathrm{GAP})), \tag{5.5}$$

where GAP is global average pooling with the scale of $1 \times 1$ and produces a vector with the size of 512, $f_c$ is a fully connected layer, and Sig is the sigmoid function that yields the value in a range of [0,1] to find the impact of each layer in the feature maps.

Thus, the classification module produces a class probability, which is a value in the range of [0,1]. If the output probability ($\mathrm{Cl}_t^{est}$) is less than 0.5, the model considers the input as a non-dense crowd image and passes it to the sparse DAD branch. Otherwise, it passes it to the high DAD branch, as shown in Fig. 5.2.

### 5.3.5 Density Aware Decoder (DAD)

DAD is one of the special modules of our proposed PDANet model, as it dynamically handles intra-variation of the density level within the input image. To achieve this, we use four dilated convolution layers with the attention module attached to

each layer, similar to the one introduced in Section 5.3.2. According to the result of the classification module, we pass the output of the PFE module ($F^p_{out}$) to one of two DAD modules. If the input scene is highly crowded, we direct $F^p_{out}$ to the high DAD branch. Otherwise, we pass it to the sparse branch. We achieve a model that can address the density variation of among different input image adaptively. Furthermore, the DAD module by itself is composed of two parts, *i.e.*, the shared layers, and the low or high-density decoder branches. This design enables us to cope with various occlusions, internal changes, and diversified crowd distributions within every single input scene, as illustrated in Fig. 5.1.

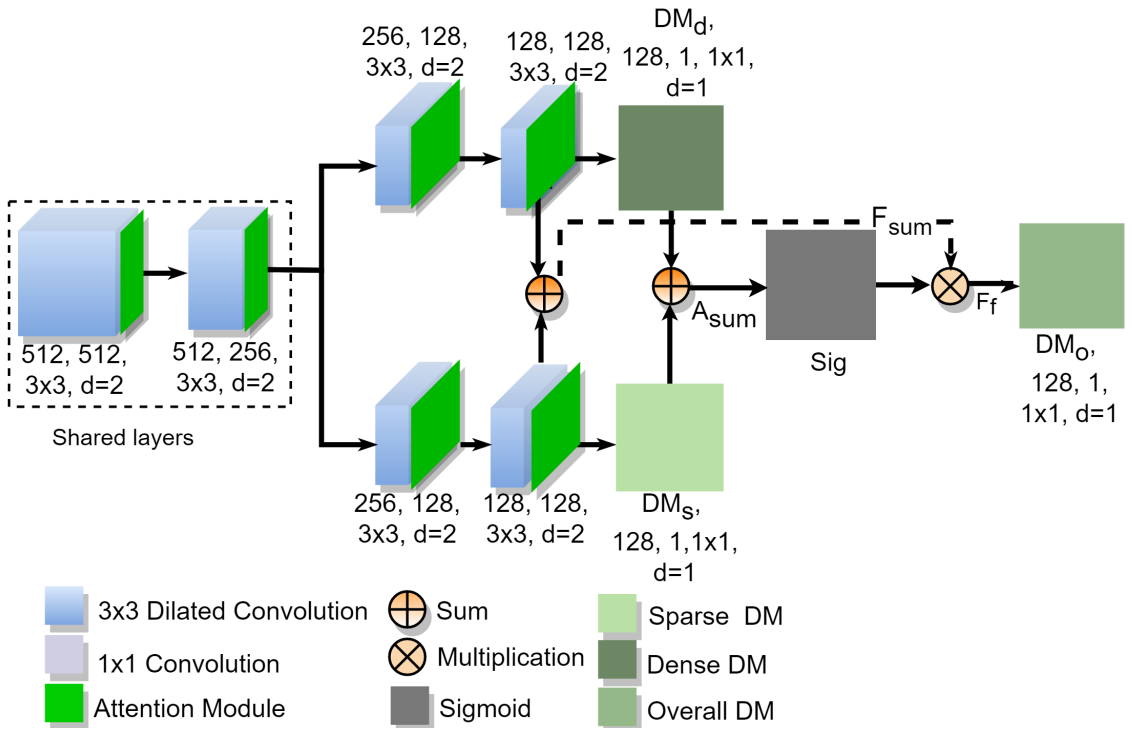



Figure 5.6 : **The illustration of the DAD module. The input feature maps are fed to the two shared layers and then we use the two branches with three convolution layers to handle the dense and sparse areas within the scene.**

The structure of DAD is illustrated in Fig. 5.6. As shown in the figure, we

consider the first two layers as shared layers and then pass the output feature map along two separate paths with the other three convolution layers to manage the within-image density variation, as shown in Fig. 5.1. The number of channels in the dilated convolution in DAD is ($N_{ch} = 512, 256, 128, 128, 1$) with the kernel filter size $3 \times 3$ and the dilation rate $d_{rate} = 2$ for the first four layers and $1 \times 1$ convolution at the end to produce the density maps. We call the output of dense and sparse branches as $\text{DM}_d$ and $\text{DM}_s$, respectively. Furthermore, to reduce the number of training parameters, we utilise a $1 \times 1$ convolution to reduce the input channels to 32 and then perform a 2D dilated convolution on the reduced channel feature maps. This process speeds up the training and convergence of our model.

Moreover, there is a small notation for the dense and non-dense crowded areas. We use the $\text{DM}_d$ for the high dense regions within the image. However, for the low density regions, within a low or highly dense input image, we use a shared $\text{DM}_s$ layer. This design gives us the benefit of using more information to train the model to map the low and dense regions with the input image. Therefore, we are able to have a better density estimation for the low crowded areas. On the other hand, by utilizing a different $\text{DM}_d$ for the highly dense areas within the input image, our DAD module is able to improve its estimation for these areas too.

By utilizing this architecture in the DAD, we will have two resultant density maps for the low and high crowded areas of the input image. Besides this, we pick up these feature maps of the last layer in the dense and non-dense branches. Then, we sum up these feature maps to form an attention module $A_{\text{sum}}$, and name the summation as $F_{\text{sum}}$. Therefore, we use the following equation to produce the final overall feature map:

$$F_f = F_{sum} \times \ \text{Sig}(A_{\text{sum}}), \tag{5.6}$$

where $\text{Sig}(A_{\text{sum}})$ is the sigmoid scaling of $A_{\text{sum}}$, and $F_f$ is the final overall feature map, which is fed to the final layer to produce an overall dense map.

### 5.3.6 Implementation Details

The last part of PDANet is about the loss function. The PDANet uses two significant losses, which fall into two categories, *i.e.*, the regression and the classification losses. We explain them in details in the following sections.

### 5.3.7 Regression Loss and Ground Truth

For the regression loss, we utilise a combination of three different error measurements, i.e., counting error $\ell^c$, various scale error $\ell^2$, and escalated error $\ell^{es}$, respectively. We measure the counting error $\ell^c$ as an absolute difference between ground-truth and the estimated crowd count, with the following equation:

$$\ell^c = \left| \sum_{i=1}^{N} D_i^{gt} - \sum_{i=1}^{N} D_i^{est} \right|, \tag{5.7}$$

where $N$ is the number of pixels in an input scene, $D_i^{gt}$, and $D_i^{est}$ are the ground-truth and the estimated crowd count at location $i$ for $i = 1, 2, \cdots, N$, respectively.

We rely on the same methodology as that in the previous work to obtain the ground-truth density map $D_i^{gt}$ [9], which is generated by convolving each delta function $\delta(x - x_i)$ with a normalized Gaussian kernel $G_\sigma$ [9] as:

$$D_i^{gt} = \sum_{x_\in S_I} \delta(x - x_i) \times G_\sigma(x), \tag{5.8}$$

where $S_I$ represents the number of annotated points in the image $I$, and $x_i$ is the $i$-th annotated point.

Note summation of the density maps ($D_i^{gt}$, $i = 1, 2, \cdots, N$) is equal to the crowd count in the image. Instead of using the geometry-adaptive kernels [76] in Eq. 5.8, we use a fixed spread parameter $\sigma$ of the Gaussian kernel for generating ground truth density maps.

For the proposed PDANet, we need to separate sparse and dense regions within the input scene to extract the dense and sparse regression losses for each input

image. To obtain $D^{gt}_{d,i}$ or $D^{gt}_{s,i}$ representing the density map at location $i$ falling into a dense or sparse region in the input image, we utilise a simple rule, which is defined by:

$$D^{gt}_{d,i} = \begin{cases} \mathrm{D}^{gt}_i, & \text{if } D^{gt}_i > mean\{D^{gt}_i, i = 1, 2, \cdots, N\}, \\ 0, & \text{else.} \end{cases} \tag{5.9}$$

$$D^{gt}_{s,i} = \begin{cases} \mathrm{D}^{gt}_i, & \text{if } D^{gt}_i <= mean\{D^{gt}_i, i = 1, 2, \cdots N\}, \\ 0, & \text{else.} \end{cases} \tag{5.10}$$

The various scale error $\ell^2$ measures the pixel-wise errors of various scales. To obtain this error, we again utilise the Global Average Pooling (GAP) on the ground truth and the estimated density map. We apply three scales of GAP, i.e., dividing each of the target and input density maps with divisors of 2, 4 and 8, respectively. For example, for the input density map with size of $H \times W$, we apply GAP with a divisor of 2 to generate the corresponding GAPs' density maps with the size of $\frac{H}{2} \times \frac{W}{2}$. Then, we measure the $\ell^2$ for each scale by:

$$\ell^2 = \sum_{i=1}^{N} \left| D^{gt}_i - D^{est}_i \right|^2, \tag{5.11}$$

where, for each $i \in \{1, 2, \cdots, N\}$, we use the same $D^{gt}_i$ and $D^{est}_i$ to represent the ground-truth and the estimated maps for each of the three different scales, respectively. We denote these three scale errors $\ell^2_{s2}$, $\ell^2_{s4}$ and $\ell^2_{s8}$, respectively. These errors help us to accurately handle the density variation in each scale of the input scene.

The escalated error mostly focuses on addressing the area with a high difference between the ground truth and the estimated density map. To extract this error, we need to extract the absolute difference between the estimated density map and its ground truth density at each location $i$ as as shown in Eq. 5.12 below, where $i = 1, 2, \cdots, N$.

$$\ell^1_i = \left| D^{gt}_i - D^{est}_i \right|. \tag{5.12}$$

Then, we calculate the average difference $\ell^1$ from $\ell_i^1$, $i = 1, 2, \cdots, N$, by:

$$\ell^1 = \frac{\sum_{i=1}^{N} \ell_i^1}{N}. \tag{5.13}$$

Then, we use $\ell^1$ to add extra weight to the area with higher misestimated value to speed up the training process. We also force PDANet to generate escalated error values for a region with no people or objects. It is done by augmenting the corresponding $\ell_i^1$ values by 10 times. The escalated difference error $\ell_i^{es}$ at location $i$, for $i = 1, 2, \cdots, N$, and the overall escalated error $\ell^{es}$ are defined by:

$$\ell_i^{es} = \begin{cases} 10 \times \ell_i^1, & \text{if } \ell_i^1 <= \ell^1, \\ 10 \times \ell_i^1, & \text{else if } D_i^{gt} == 0, \\ \ell_i^1, & \text{else,} \end{cases} \tag{5.14}$$

and,

$$\ell^{es} = \sum_{i=1}^{N} \ell_i^{es}. \tag{5.15}$$

Then, the overall regression loss, denoted by $\ell_o^{reg}$, is defined by:

$$\ell_o^{reg} = \ell^{es} + \ell^c + \ell_{s2}^2 + \ell_{s4}^2 + \ell_{s8}^2. \tag{5.16}$$

When $D_i^{gt}$ in Eqs. 5.7, 5.11, and 5.12, is replaced by $D_{d,i}^{gt}$ and $D_{s,i}^{gt}$, respectively, then the results from Eq. 5.16 define the dense regression loss and the sparse regression, which are denoted by $\ell_d^{reg}$ and $\ell_s^{reg}$, respectively.

### *Classification Loss and Ground Truth*

On the other hand, according to our model, we need to classify the scene. Thus, we introduce $\mathrm{Cl}_t^{gt}$ as an actual class tag. To obtain the $\mathrm{Cl}_t^{gt}$, we define a rule to decide whether the input image is highly crowded or not.

We consider $Dl_p^{gt}$ as a measure of ground truth dense level of the input scene, which is defined by:

$$Dl_p^{gt} = \frac{\sum_{i=1}^{N} D_i^{gt}}{\sum_{i=1}^{N} \mathrm{sgn}(D_i^{gt})}, \tag{5.17}$$

where sgn$(\cdot)$ is a sign function.

Then, according to the changes in the number of people in each dataset, we can find a threshold $\tau$, thus, we define a $\mathrm{Cl}_t^{gt}$ by:

$$Cl_t^{gt} = \begin{cases} 1, & \text{if } Dl_p^{gt} > \tau, \\ 0, & \text{else.} \end{cases} \tag{5.18}$$

If the dense level of the input scene $Dl_p^{gt}$ is larger than the threshold $\tau$, we consider it as a high density input scene; otherwise, it is a low density one. We test different threshold values of $\tau$, and found that our model is not too sensitive to it and able to classify the input scenes correctly. On the other hand, we can obtain the $\mathrm{Cl}_t^{est}$ from the classification module in Eq. 5.5.

Then, we consider the Binary Cross Entropy (BCE) loss to train the model to detect sparse and dense input images, where $BCE_{loss}$ is defined by:

$$BCE_{loss} = -\left[\mathrm{Cl}_t^{gt} \cdot \log \mathrm{Cl}_t^{est} + (1 - \mathrm{Cl}_t^{gt}) \cdot \log(1 - \mathrm{Cl}_t^{est})\right]. \tag{5.19}$$

***Total Loss***

Finally, we need to define a rule to train the model efficiently by a combination of proposed losses. As it is obvious from the structure of the model, we need to detect and correctly pass high and sparse dense input to the corresponding DAD. Therefore, we need to penalize the model whenever it cannot detect the dense level of the input scene. Thus, we use the following equation to combine different losses:

$$\mathrm{Sum}_{loss} = \ell_o^{reg} + \alpha \times (\ell_d^{reg} + \ell_s^{reg}), \tag{5.20}$$

and

$$\mathrm{Final}_{loss} = BCE_{loss} \times \ell_o^2 + \mathrm{Sum}_{loss}, \tag{5.21}$$

where $\alpha$ is set to 0.4 according to empirical studies.

According to the $\text{Final}_{loss}$, by adding the $BCE_{loss} \times \ell_o^2$, we are able to overcome the mis-classification of the input scene. With $\text{Sum}_{loss}$, the model can learn the dense and sparse area within an input image precisely.

## 5.4 Experiments

In this section, we evaluate the performance of our proposed approach. To investigate the performance of the proposed model, we use the MAE and MSE metrics that we defined in the section 2.4, by equations 2.1 and 2.2. The experiments are conducted on four benchmark datasets, and results are compared with the recently published state-of-the-art approaches, which have already been used for comparison purpose since.

### 5.4.1 Data Augmentation

We take the benefit of data augmentation to avoid the risk of over-fitting to the small number of training images. We use five types of cropping alongside with a resizing as data augmentations. We crop each image into $\frac{1}{4}$ of the original dimension. The first four cropped images extract four non-overlapping patches based on each corner of the original image. Furthermore, the fifth crop is randomly cropped from the input scene. For resizing, we just resize the input image to the dimension of $(768, 1024)$ or $(1024, 768)$ depending on the scale of the input data. If the height of an input image is bigger than the width of it, we just select $(1024, 768)$, and in other case we resize it to $(768, 1024)$ size.

### 5.4.2 Experimental Results on the Shanghai Tech Dataset

As the challenge caused by diversity of scenarios and variation of congestion differs on the Shanghai Tech dataset [183], it is difficult to estimate the number of pedestrians precisely. Following [9] and as mentioned in Section 2.4, for setting $\sigma$

Table 5.1 : **Comparison of the MAE and MSE results obtained with our proposed PDANet and the-state-of-the-art crowd counting approaches on the Shanghai Tech part A dataset [183].**

| Methods | MAE | MSE |
|---|---|---|
| ACSCP [128] | 75.7 | 102.7 |
| D-ConvNet-v1 [178] | 73.5 | 112.3 |
| IG-CNN [5] | 72.5 | 118.2 |
| GWTA-CCNN [122] | 154 | 229 |
| DRSAN [85] | 69.3 | 96.4 |
| ic-CNN [111] | 68.5 | 116.2 |
| CSRNet [76] | 68.2 | 115.0 |
| SANet [9] | 67.0 | 104.5 |
| DENet [83] | 65.5 | 101.2 |
| SFCN [156] | 64.8 | 107.5 |
| TEDnet [31] | 64.2 | 109.1 |
| ADCrowdNet [86] | 63.2 | 98.9 |
| PACNN+CSRNet [130] | 62.4 | 102.0 |
| CAN [88] | 62.3 | 100.0 |
| HA-CCN [138] | 62.9 | 94.9 |
| SPN [15] | 61.7 | 99.5 |
| **PDANet** | **60.8** | **93.4** |

Table 5.2 : **Comparison of the MAE and MSE results obtained with our proposed PDANet and the-state-of-the-art crowd counting approaches on the Shanghai Tech part B dataset [183].**

| Methods | MAE | MSE |
|---|---|---|
| ACSCP [128] | 17.2 | 27.4 |
| D-ConvNet-v1 [178] | 18.7 | 26.0 |
| IG-CNN [5] | 13.6 | 21.1 |
| DecideNet [81] | 21.53 | 31.98 |
| DRSAN [85] | 11.1 | 18.2 |
| ic-CNN [111] | 10.7 | 16.0 |
| CSRNet [76] | 10.6 | 16.0 |
| SANet [9] | 8.4 | 13.6 |
| DENet [83] | 9.6 | 15.4 |
| SFCN [156] | 7.6 | 13.0 |
| TEDnet [31] | 8.2 | 12.8 |
| ADCrowdNet [86] | 7.7 | 12.9 |
| PACNN [130] | 8.9 | 13.5 |
| CAN [88] | 7.8 | 12.2 |
| HA-CCN [138] | 8.1 | 13.4 |
| SPN [15] | 9.4 | 14.4 |
| **PDANet** | **7.1** | **10.9** |

for part A, we use the KNN method to calculate the average distance between each head and its three nearest heads and $\beta$ is set to 0.3. For part B, we set a fixed value 15 for $\sigma$. We compare our method with state-of-the-art methods recently published on this dataset.

The quantitative results for Shanghai Tech-A are listed in Table 5.1. We collect results of the state-of-the-art approaches from their original published papers. It can be seen that our PDANet has achieved an MAE of 58.5 and an MSE of 93.4 in the experiment. Our proposed method also exhibits significant advantages over many top ranked methods such as PaDNet [146], ADCrowdNet [86], HA_CNN [138], and SPN [15]. On the Shanghai Tech-B dataset, As shown in Table 5.2 our proposed PDANet has achieved an MAE of 7.1 and an MSE of 10.9, both are better than those of the state-of-the-art results. These results suggest that our proposed PDANet is able to cope with sparse and dense scenes, thanks to the combination of the pyramid module as mentioned in Section 5.3.3 and the two-branch DAD as described in Section 5.3.5. Because of these, our proposed model can distinguish the crowd level of the input scene and analyze the crowd accordingly for better estimation.

### 5.4.3 Experimental Results on the WorldExpo10 Dataset

Table 5.3 also provides MAE results based on five different scenes on the World-Expo10 dataset [176]. The best-performing state-of-the-art methods are CAN [88], ADCrowdNet [86], and PACNN [130] with an average MAE less than 8. However, as shown in the table, our proposed PDANet has achieved an average MAE of 6.0, which suppresses the-state-of-the-art results with a margin of 1.4 over the results achieved by CAN [88]. Furthermore, our PDANet yields the lowest MAE of 4 out of all 5 scenes with an MAE values equal to 1.8, 9.1, 7.3, and 2.2, respectively. As it is demonstrated, the overall performance of our PDANet across various scenes is superior compared with the-state-of-the-art approaches.

Table 5.3 : **Comparison of the MAE results obtained with our proposed PDANet and the-state-of-the-art crowd** counting approaches on the World-Expo10 dataset [176].

| Methods | Sce.1 | Sce.2 | Sce.3 | Sce.4 | Sce.5 | AVG |
|---|---|---|---|---|---|---|
| ACSCP [128] | 2.8 | 14.05 | 9.6 | 8.1 | 2.9 | 7.5 |
| D-ConvNet-v1 [178] | 1.9 | 12.1 | 20.7 | 8.3 | 2.6 | 9.1 |
| IG-CNN [5] | 2.6 | 16.1 | 10.15 | 20.2 | 7.6 | 11.3 |
| CP-CNN [136] | 2.9 | 14.7 | 10.5 | 10.4 | 5.8 | 8.86 |
| DRSAN [85] | 2.6 | 11.8 | 10.3 | 10.4 | 3.7 | 7.76 |
| ic-CNN [111] | 17.0 | 12.3 | 9.2 | 8.1 | 4.7 | 10.3 |
| CSRNet [76] | 2.9 | 11.5 | 8.6 | 16.6 | 3.4 | 8.6 |
| SANet [9] | 2.6 | 13.2 | 9.0 | 13.3 | 3.0 | 8.2 |
| DENet [83] | 2.8 | 10.7 | 8.6 | 15.2 | 3.5 | 8.2 |
| DecideNet [81] | 2.0 | 13.14 | 8.9 | 17.4 | 4.75 | 9.23 |
| TEDnet [31] | 2.3 | 10.1 | 11.3 | 13.8 | 2.6 | 8.0 |
| ADCrowdNet [86] | 1.7 | 14.4 | 11.5 | 7.9 | 3.0 | 7.7 |
| PACNN [130] | 2.3 | 12.5 | 9.1 | 11.2 | 3.8 | 7.8 |
| CAN [88] | 2.9 | 12.0 | 10.0 | 7.9 | 4.3 | 7.4 |
| BSAD [123] | 4.1 | 21.7 | 11.9 | 11.0 | 3.5 | 10.5 |
| SaCNN [107] | 2.6 | 13.5 | 10.6 | 12.5 | 3.3 | 8.5 |
| **PDANet** | **1.8** | **9.1** | 9.6 | **7.3** | **2.2** | **6.0** |

### 5.4.4 Experimental Results on the UCF Dataset

We choose the setting similar to the Shanghai Tech-A [183] setting for generating ground truth density maps on the UCF CC 50 dataset [55]. Table 5.4 shows that on

Table 5.4 : **Comparison of the MAE and MSE results obtained with our proposed PDANet and state-of-the-art crowd counting approaches on the UCF crowdcounting dataset [55].**

| Methods | MAE | MSE |
|---|---|---|
| A-CCNN [2] | 367.3 | 423.7 |
| ACSCP [128] | 291.0 | 404.6 |
| D-ConvNet-v1 [178] | 288.4 | 404.7 |
| IG-CNN [5] | 291.4 | 349.4 |
| ASD [162] | 196.2 | 270.9 |
| DRSAN [85] | 219.2 | 250.2 |
| ic-CNN [111] | 260.9 | 365.5 |
| CSRNet[76] | 266.1 | 397.5 |
| SANet [9] | 258.4 | 334.9 |
| DENet [83] | 241.9 | 345.4 |
| SFCN [156] | 214.2 | 318.2 |
| TEDnet [61] | 249.4 | 354.5 |
| ADCrowdNet [86] | 257.1 | 363.5 |
| PACNN [130] | 267.9 | 357.8 |
| CAN [88] | 212.2 | 243.7 |
| HA-CCN [138] | 256.2 | 348.4 |
| SPN [15] | 259.2 | 335.9 |
| SPN+L2SM [166] | 188.4 | 315.3 |
| **PDANet** | **119.8** | **159** |

this dataset our PDANet outperforms the state-of-the-art models by a significant margin. We achieve an MAE of 119.8 with an MSE of 159, which is about 35 percent better than PaDNet [146], the best-performing benchmark model. In our experiments, we observe that our PDANet is able to estimate the number of people accurately in all subsets. Overall, it can be concluded that our proposed PDANet can work well on both sparse and dense scenarios. We also explore the results in detail at the Ablation Study section.

### 5.4.5 Experimental Results on the UCSD Dataset

Table 5.5 shows the MAE and MSE results obtained on the UCSD dataset [11]. Compared with nine currently best approaches tested on this low crowed dataset, the proposed PDANet achieves the second best results with an MAE of .93 and an MSE of 1.21, which are very close and very comparative to the best results from the PaDNet model. We believe that the results is good enough taking into account that extreme resizing (image size in the UCSD dataset is $238 \times 158$) is needed for PDANet as we mentioned in Sec. 5.4.1 to work in our model.

## 5.5 Ablation Study

To further demonstrate the effectiveness of each component proposed in our PDANet model, we conduct a series of ablation studies.

In this section, we first visualize some examples of the results achieved, and then explore some of our model components and discuss their outputs to analyze the effectiveness of each component. The ablation studies are conducted on the UCF CC50 [55] and the Shanghai Tech [183] datasets.

Table 5.5 : **Comparison of the MAE and MSE results obtained with our proposed PDANet and the-state-of-the-art crowd counting approaches on the UCSD crowd-counting dataset [11].**

| Methods | MAE | MSE |
|---|---|---|
| Density Learning [71] | 1.70 | 1.28 |
| Learning to Count[31] | 1.70 | 2.16 |
| Count Forest [109] | 1.43 | 1.30 |
| Arteta et al. [4] | **1.24** | 1.31 |
| Zhang et al. [176] | 1.70 | 1.2 |
| Switch-CNN [123] | 1.62 | 2.10 |
| ConvLSTM [163] | 1.30 | 1.79 |
| A-CCNN [2] | 1.51 | 1.36 |
| Bidirectional ConvLSTM [163] | 1.13 | 1.43 |
| CSRNet[76] | 1.16 | 1.47 |
| ACSCP [128] | 1.04 | 1.35 |
| SANet [9] | 1.02 | 1.29 |
| BSAD [54] | 1.00 | 1.40 |
| SPN [15] | 1.03 | 1.32 |
| ADCrowdNet(DME) [86] | 0.98 | 1.25 |
| PACNN [130] | **0.89** | **1.18** |
| **PDANet** | 0.93 | 1.21 |

### 5.5.1 Density Map Visualization

Qualitatively, we visualize the density maps generated by our proposed PDANet method on the Shanghai Tech part A, part B [183], and UCF CC 50 [55] datasets in

comparison with the original ground truth (GT). These are shown in Figs. 5.7, 5.8 and 5.9. In these figures, three sample images corresponding to low, medium and high density scenes are selected from each dataset. For each sample image, we show the input image with an index of 0, and its Ground Truth (GT) density map, its estimated overall density map, and its estimated dense and sparse density maps with an index of 1 to 4, respectively.

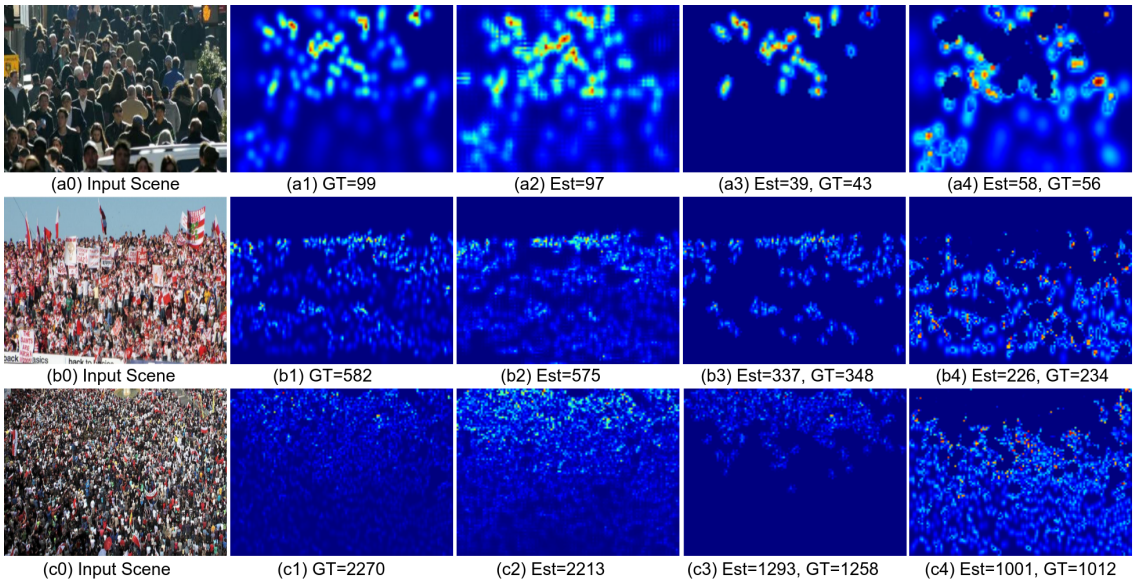



Figure 5.7 : **Results of the estimated density maps of images from the Shanghai Tech part A dataset. We illustrate three test images (a0, b0, c0), their actual ground truth (a1, b1, c1), our estimated overall density maps (a2, b2, c2), our estimated density maps for dense areas (a3, b3, c3), and our estimated density maps for sparse areas and their crowd counts (a4, b4, c4).**

Fig. 5.7 presents some sample results obtained on the Shanghai Tech part A dataset. For this dataset, we select three images with a total crowd count of 99, 582 and 2,270 respectively, representing input scenes of low (top row of Fig. 5.7), medium (middle row of Fig. 5.7) and high (bottom row of Fig. 5.7) crowdedness

scenes.

As shown in this figure, the estimated counts and the actual ground truth counts are very close to each other, demonstrating that our proposed model performs well in the scenes of various crowdedness levels. For instance, for the image in the bottom row of Fig. 5.7, the ground truth count is 2,270, while our prediction is 2,213, which is a reasonable estimation for such a highly crowded input scene. On the other hand, for low crowdedness scenes, such as Fig. 5.7(a0), our proposed PDANet also produces accurate density maps. Fig. 5.7 also shows that our proposed model can accurately discriminate more crowded areas from less crowded ones. When looking further into the results of dense and sparse scenes, we can draw a conclusion that our model works well for extracting better information for more accurate overall density map estimation.

Fig. 5.8 presents results on three sample scenes from the Shanghai Tech part B dataset. In this figure, we choose three sample images with crowd counts varying from 29 to 251, corresponding to low, medium, and high crowdedness images. These figures also demonstrate that our PDANet works well in low crowdedness areas. For instance, in Fig. 5.8(a0), the predicted density map and the actual density map appear to be very similar, and so are the estimated count and the ground truth count of crowd in the scene. Fig. 5.8 also shows that for medium and low crowdedness scenes, our proposed model produces accurate density maps.

Fig. 5.9 illustrates the results on three sample images from the UCF CC 50 dataset [55], which is a highly crowded challenging dataset. In this figure, we choose three images with crowd counts equal to 555, 1,852, and 4,706, corresponding to low, medium, and high crowdedness scenes, respectively. We can see that our proposed PDANet works well in highly crowded images as well as low and medium crowdedness images. It is also evident that our proposed DAD model helps to localize dense

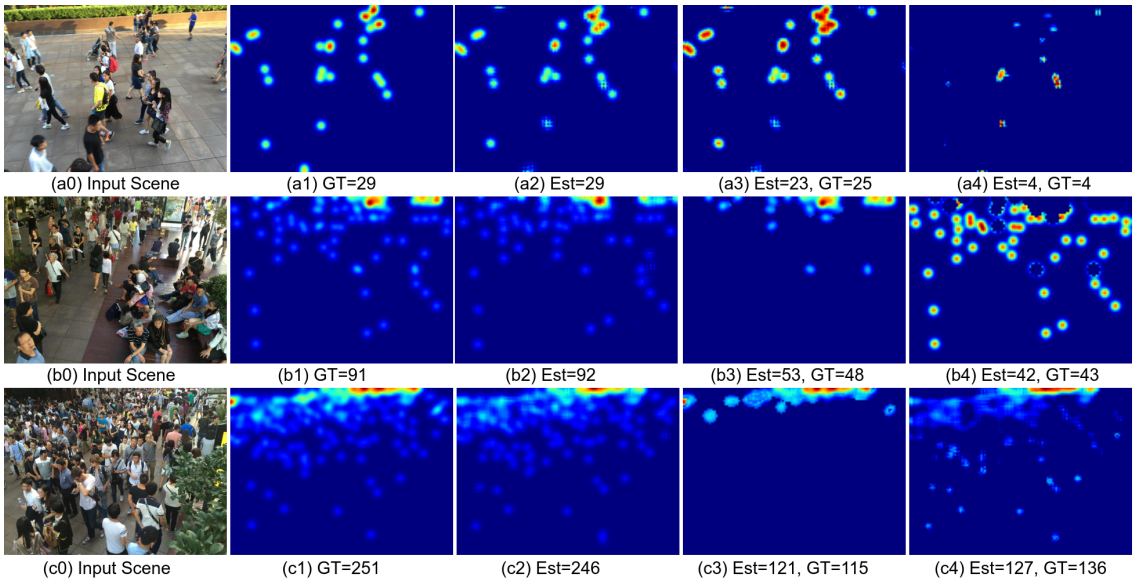



Figure 5.8 : **Results of the estimated density maps of images from the Shanghai Tech part B dataset. We illustrate three test images (a0, b0, c0), their actual ground truth (a1, b1, c1), our estimated overall density maps (a2, b2, c2), our estimated density maps for dense areas (a3, b3, c3), and our estimated density maps for sparse areas and their crowd counts (a4, b4, c4).**

and non-dense areas of the input image. However, in the medium crowedness images, it is evident that some plant areas are considered as crowd, due to the nature of the grayscale input image.

### 5.5.2 Effectiveness of the PFE Module

In the first experiment, we investigate the impact of different numbers of GAP modules on the baseline model (baselineAD, *i.e.*, a PDANet without the PFE module).

We test our proposed model with different numbers of GAPs from 0 GAP (baselineAD) to 10 GAPs. We obtain the GAPs of input feature maps by resizing them

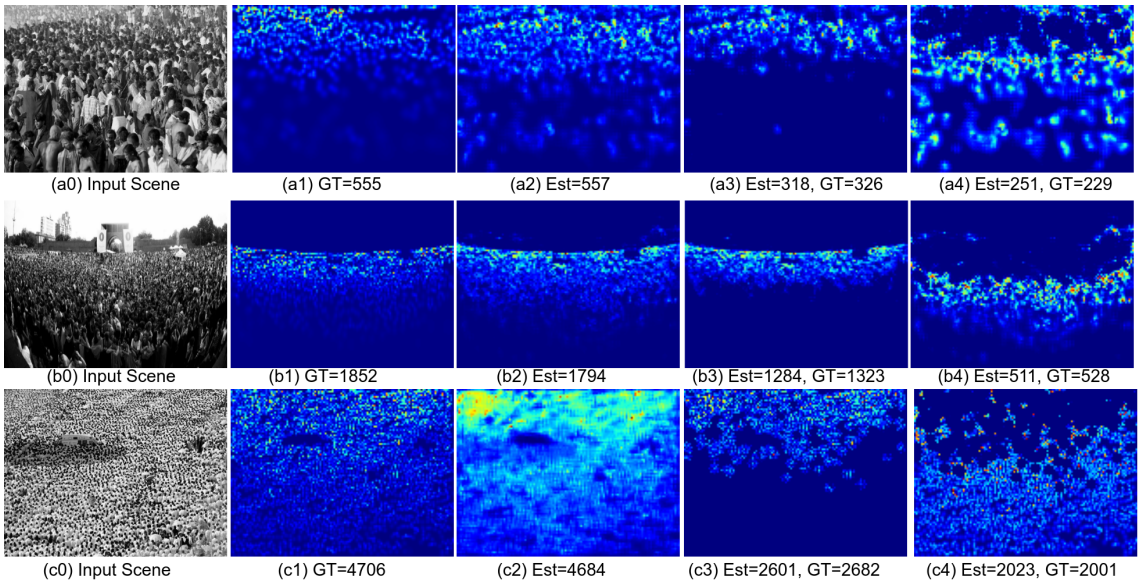



Figure 5.9 : **Results of the estimated density maps of images from the UCF CC 50 dataset [55]. We present three test images (a0, b0, c0), their actual ground truth (a1, b1, c1), our estimated overall density maps (a2, b2, c2), our estimated density maps for dense areas (a3, b3, c3), and our estimated density maps for sparse areas and their crowd counts (a4, b4, c4).**

with divisors of 2, 4, 8, 3, 6, 10, 5, 7, and 9, respectively. For example, for the input feature map with size of $H \times W$, we apply GAP with divisor of 2 to generate the corresponding GAPs' feature maps with the size of $\frac{H}{2} \times \frac{W}{2}$. We sort these numbers with the order of use. For example, if we aim to utilise 3 GAPs, we divide the input feature maps with the divisors of 2, 4, ..., respectively, to capture the information of various scales from the input feature maps.

Fig. 5.10 presents the results of this experiment on part0 of the UCF CC 50 dataset. In this figure, we report the achieved MAE and MSE results for the PFE module with various GAPs. As shown in this graph, our PDANet has achieved an MAE of 157 and an MSE of 202 at three GAP settings (the third one in Fig. 5.10),

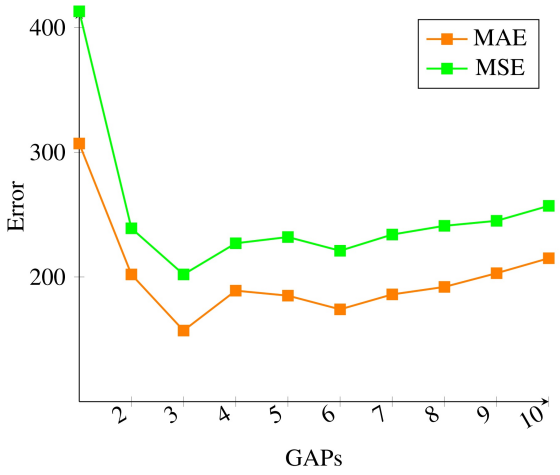



Figure 5.10 : **Comparison of MAE and MSE results between various numbers of GAP layers on the UCF CC 50 crowd counting [55].**

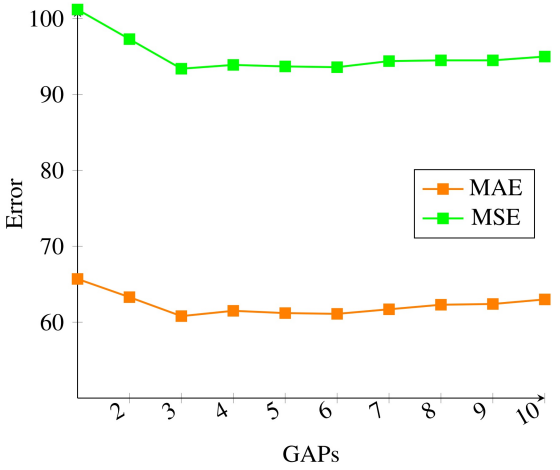



Figure 5.11 : **Comparison of MAE and MSE results between various numbers of GAP layers on the Shanghai Tech partA [183].**

utilizing the division factors 2, 4, and 8, as three different scales of input feature maps. As it is shown, the proposed PDANet with this setting outperforms other PDANets with more or fewer GAPs modules, as well as the baselineAD model. Among the various PFE modules, PFEs with three GAPs (3GAP) and six GAPs (6GAP) provide better crowd level predictions in part0. Fig. 5.10 also shows that the PDANet with GAPs in the worst-case still improves the estimation of the baselineAD

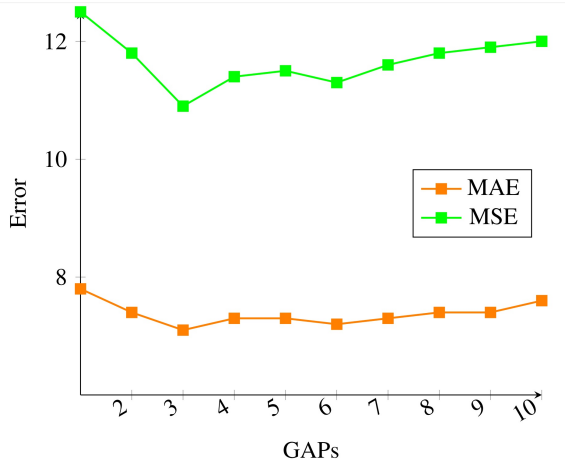



Figure 5.12 : **Comparison of MAE and MSE results between various numbers of GAP layers on the Shanghai Tech part B [183].**

(MAE of 202 vs. 300).

We test the effect of different numbers of pyramid GAPs on the Shanghai Tech dataset as well. The test results are shown in Fig. 5.11 and Fig. 5.12, for the Shanghai Tech part A and part B, respectively. The results again show that our proposed PDANet (with three GAPs) outperforms the baselineAD and other PDANet models with more or fewer GAPs. The results of the other numbers of GAPs fluctuate slightly from an MAE of 58.5 to 65.7 for part A and an MAE of 7.3 to 7.8 for part B, which are very consistent. In summary, the PFE with three GAPs works better in the PDANet model for crowd counting. We believe that by using the scale like the one used in PDANet (i.e., three GAPs or 3GAP), the output feature maps have more accurate scale information than those of the other PDANet models with different numbers of GAPs. On the other hand, increasing the number of GAPs will increase the number of parameters, which in turn increases the complexity of the model. Thus, the performance of the model will slightly decrease with the rise of over-fitting issues. Overall, our proposed PDANet (3GAP) has the most optimal number of parameters for the PFE modules, too. Thus, it can be trained more

efficiently by the model to capture the essential scale information.

### 5.5.3 Effectiveness of the Attention Module

To gain an insight into the effectiveness of the Attention Module, we perform an ablation study to demonstrate the contribution of the module to the performance of the proposed model. We compare the performance of our design choices with the baseline with PFE and DAD module. Tables 5.6 and 5.7 illustrate the results obtained on the UCF CC 50 and the Shanghai Tech datasets. part0 of UCF CC 50 dataset has the greatest improvement in terms of MAE/MSE, but the improvement on the performance of part1 to part4 is small. As shown in Table 5.7, we achieved more or less the same improvement in crowd counting by adopting the attention module.

Overall, we use the attention module for localizing the crowd area and improving the performance of our model. As shown in these tables, we achieve our goal by combining spatial/channel based attentions on both sparsely and densely crowded areas. Thus, these results prove the application of the attention module on improving the accuracy of the crowd counting model.

### 5.5.4 Effectiveness of the Classification and DAD Modules

To address the density variation within and between different input images, we proposed a two-branch DAD module. In this section, we aim to understand the effect of this module in our overall performance improvement. Like what were done in the previous sections, we compare the results of our PDANet with DAD and without DAD (passing the data to one branch only) on both UCF CC 50 and the Shanghai Tech datasets.

Tables 5.8 and 5.9 show the experimental results obtained on the UCF CC 50 and the Shanghai Tech datasets, respectively. As seen from Table 5.8, we are able to

Table 5.6 : **Effect of adopting the attention module on crowd counting performance based on the UCF crowd-counting dataset [55].**

| | UCF CC 50 | | | | | | |
|---|---|---|---|---|---|---|---|
| | Metrics | part0 | part1 | part2 | part3 | part4 | AVG |
| BaselinePD | MAE | 205 | 132 | 86 | 127 | 112 | 132.6 |
| | MSE | 243 | 164 | 111 | 188 | 131 | 167.4 |
| PDANet | MAE | **157** | **128** | **80** | **126** | **108** | **119.8** |
| | MSE | **202** | **182** | **95** | **186** | **130** | **159** |

Table 5.7 : **Effect of Attention Module on crowd counting performance based on the Shanghai Tech crowd-counting dataset [183].**

| | Metrics | Shanghai Tech | |
|---|---|---|---|
| | | part A | part B |
| BaselinePD | MAE | 62.3 | 7.3 |
| | MSE | 98.6 | 11.6 |
| PDANet | MAE | **58.5** | **7.1** |
| | MSE | **93.4** | **10.9** |

Table 5.8 : **Effect of classification and DAD modules on crowd counting performance based on the UCF crowd counting dataset [55].**

| | | UCF CC50 | | | | | |
|---|---|---|---|---|---|---|---|
| | Metrics | part0 | part1 | part2 | part3 | part4 | AVG |
| BaselinePA | MAE | 217 | 151 | 116 | 146 | 114 | 148.8 |
| | MSE | 267 | 183 | 124 | 185 | 138 | 179.4 |
| PDANet | MAE | **157** | **128** | **80** | **126** | **108** | **119.8** |
| | MSE | **202** | **182** | **95** | **186** | **130** | **159** |

boost the accuracy of crowd counting by about 20 percent for the UCF dataset in all subsets. With the Shanghai Tech dataset, we also achieve a noticeable improvement in accuracy with the help of the DAD module.

Table 5.9 : **Effectiveness of the classification and DAD modules on crowd counting performance based on the Shanghai Tech crowd-counting dataset [183].**

| | Metrics | Shanghai | |
|---|---|---|---|
| | | partA | partB |
| BaselinePA | MAE | 66.5 | 7.5 |
| | MSE | 104.1 | 12.6 |
| PDANet | MAE | **58.5** | **7.1** |
| | MSE | **93.4** | **10.9** |

These results demonstrate the effectiveness of our initial idea about processing the sparsely and densely crowded feature maps separately. We believe that the DAD module helps the PDANet generate proper density maps for both high and low crowdedness areas in the images, and simultaneously, it guides the proposed model to react to the difference among input images with different crowdedness.

## 5.6 Conclusion

In this work, we have introduced a novel deep architecture called Pyramid Density-Aware Attention-based network (PDANet) for crowd counting. The PDANet has incorporated pyramid features and attention modules with a density-aware decoder to address the huge density variation within the crowded scenes. The proposed PDANet has utilised a classification module for passing the pyramid features to the most suitable decoder branch to provide more accurate crowd counting with two-scale density maps. To aggregate these density maps, we have taken the benefit of the sigmoid function and produced a gating mask for producing the final density map. Extensive experiments on various benchmark datasets have demonstrated the performance of our PDANet in terms of robustness, accuracy, and generalization. Our approach is able to achieve better performance on almost all of the major crowd counting datasets over the state-of-the-art methods, especially on the UCF CC 50 with more than 35 percent immediate improvements in the results.

# Chapter 6

# Conclusions and Future Work

This chapter summarises the thesis and presents some direction for future research. This thesis aims to explore the application of a convolutional neural network in the crowd counting research area and proposes some end-to-end solutions based on CNNs to improve the performance of crowd counting and density estimation models. Those enhancements include augmenting the effect of positive features, making CNN more adaptive, and providing end-to-end models based on CNNs for crowd counting. This study's final goal is to come up with the end-to-end automatic crowd counting and density estimation solutions. We have adopted/proposed several convolutional neural network-based approaches to building intelligent solutions for dynamic density estimation.

In the first part of the thesis, we have adopted the state-of-the-art approaches of 2017 (CCNN and MCNN) and have tried to improve their performance by some innovative ideas. At first, we introduced an adaptive hyper-parameters (HPs) optimization methods, which adaptively trained and evaluated the patch-based CCNN. This method took advantage of the fuzzy system and combine it with CNNs to find the optimal HPs for each patch in the CCNN models. The second innovation proposal was a convolutional network pruning, which enhanced the performance of crowd counting by pruning. The proposed solution enhanced the performance of the crowd counting models by removing the channels with the negative effect on the crowd counting. The experimental results proved the effectiveness of these ideas.

In the second part of the thesis, we proposed end-to-end solutions for the crowd

counting. Based on the literature review and our experience in the first part of my research, we proposed two innovative models. In the first model, we combined the detection and regression-based crowd counting and proposed a novel method for crowd counting, which fused the detected crowd's results with the estimated crowd and produced the final density maps and crowd number. The second method was an adaptive density-based solution, which addressed the intra scene and extra scenes density variations. This model combined the pyramid modules with classification and attention module to address scale and density variation challenges in crowd counting. Moreover, we defined a new combination of various losses to accelerate and improve the model's training. The experimental results proved the effectiveness of these two innovative ideas.

## 6.1 Summary of the Thesis

Chapter 2 presents a comprehensive literature review about state-of-the-art methods published related to crowd counting and density approaches. It intensely focuses on CNN-based models and verifies the taxonomy for crowd counting. This chapter classifies various models by their network architectures, such as basic CNN, multi-column, single-column. It also investigates the learning paradigm in the crowd counting solutions and provides two primary learning methods for crowd counting solutions, single-task and multi-task approaches. Moreover, chapter 2 verifies the network inference manner of crowd counting solutions (patch-based and whole image base). Also, it investigates various supervision forms in the CNN-based crowd counting solution. Besides, it provides an extensive review of the existing datasets in the crowd and object counting research area and classified them by their frequency of usage and the date they were proposed. Finally, after providing two primary evaluation metrics (MAE and MSE), it provides a comprehensive comparison based on the top six datasets in the crowd counting domain and verifies the most important

and influential features of the CN-based crowd counting solutions.

Chapter 3 presents an Adaptive CCNN architecture that takes a whole image as input and directly outputs its density map. It aims to tackle the difficult problem of crowd counting such as scale variance and extreme collusion. The proposed method has made a full use of contextual information to generate an accurate density map. To leverage the local information, it utilises the combination of CNN-based head detection and the fuzzy inference engine to choose an optimal CCNN model adaptively to each patch of the input image. Our model achieves noticeable improvements on three challenging datasets, i.e., the UCSD, UCF-CC and the crowd dataset collected by ourselves from a train station in Sydney, and have demonstrated the effectiveness of the proposed approach.

Chapter 4 proposes a new pruning strategy for crowd counting that works with CCNN and other crowd counting models. Through identifying positive, negative and irrelevant filters according to the activation of feature maps, our solution has not only reduced the network size but also improved the accuracy by removing non-contributing and negatively contributing filters. Experimental results on benchmark datasets have shown that, compared with other existing pruning algorithms, our proposed technique can improve the accuracy of counting models without fine-tuning or retraining the pruned model, and meanwhile reduce the size of the models.

Chapter 5 introduces a novel deep architecture called Pyramid Density-Aware Attention-based network (PDANet) for crowd counting. The PDANet has incorporated pyramid features and attention modules with a density-aware decoder to address the huge density variation within the crowded scenes. The proposed PDANet has utilised a classification module for passing the pyramid features to the most suitable decoder branch to provide more accurate crowd counting with two-scale density maps. To aggregate these density maps, it takes the benefit of the sigmoid

function and produces a gating mask for producing the final density map. Extensive experiments on various benchmark datasets have demonstrated the performance of our PDANet in terms of robustness, accuracy, and generalization. Our approach is able to achieve better performance on almost all of the major crowd counting datasets over the state-of-the-art methods, especially on the UCF CC 50 with more than 35% immediate improvements in the results.

## 6.2 Future Research

In the current study, we have investigated various crowd counting, and density estimation approaches to propose innovative solutions that can outperform the state-of-the-art results with good margins. Based on our experiences, several areas for further research are identified and summarised as follows.

• As I mentioned in the literature review section, a right solution for crowd counting should have a low complexity. Due to this reason, from my perspective, future research should focus more on single column based solutions.

• To address the intra dense area within a scene, it can be a good idea to use a kind of zooming technique in the middle of models, whenever a congested area recognised, to zoom on the high-density region and extract more beneficial feature from that area for accurate density estimation.

• Another extension to the crowd counting framework is patch-based processing of middle feature maps in the CNN-based model. Our initial investigation has proved that it can lead to further improvement in the accuracy of crowd counting techniques.

• To our understanding, most of the state-of-the-art crowd counting approaches, utilise the pre-trained models which were proposed before 2017. However, as we know, there are several better and faster pre-trained models that can be considered as a feature extractor to boost the accuracy and reduce the training and inference

times of crowd counting models.

# Bibliography

[1] S. Aich and I. Stavness, "Leaf counting with deep convolutional and deconvolutional networks," in *ICCV*, 2017, pp. 2080–2089.

[2] S. Amirgholipour, X. He, W. Jia, D. Wang, and M. Zeibots, "A-ccnn: Adaptive ccnn for density estimation and crowd counting," in *ICIP*. IEEE, 2018, pp. 948–952.

[3] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *ECCV*, vol. 62. Springer, 2016, pp. 483–498.

[4] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman, "Interactive object counting," in *ECCV*. Springer, 2014, pp. 504–518.

[5] D. Babu Sam, N. N. Sajjan, R. Venkatesh Babu, and M. Srinivasan, "Divide and grow: capturing huge diversity in crowd images with incrementally growing cnn," in *CVPR*, 2018, pp. 3618–3626.

[6] R. Bahmanyar, E. Vig, and P. Reinartz, "Mrcnet: Crowd counting and density map estimation in aerial and ground imagery," *BMVCW*, 2019.

[7] E. Bondi, L. Seidenari, A. D. Bagdanov, and A. Del Bimbo, "Real-time people counting from depth imagery of crowded environments," in *AVSS*. IEEE, 2014, pp. 337–342.

[8] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in *ACM MM*. ACM, 2016, pp. 640–644.

[9] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *ECCV*, 2018, pp. 734–750.

[10] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[11] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *CVPR*. IEEE, 2008, pp. 1–7.

[12] A. B. Chan and N. Vasconcelos, "Bayesian poisson regression for crowd counting," in *ICCV*. IEEE, 2009, pp. 545–551.

[13] ——, "Counting people with low-level features and bayesian regression," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2160–2177, 2011.

[14] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting." in *BMVC*, vol. 1, no. 2, 2012, p. 3.

[15] X. Chen, Y. Bin, N. Sang, and C. Gao, "Scale pyramid network for crowd counting," in *WACV*. IEEE, 2019, pp. 1941–1950.

[16] Y. Cheng, D. Wang, P. Zhou, and T. Zhang, "Model compression and acceleration for deep neural networks: The principles, progress, and challenges," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 126–136, 2018.

[17] Z.-Q. Cheng, J.-X. Li, Q. Dai, X. Wu, and A. G. Hauptmann, "Learning spatial awareness to improve crowd counting," *ICCV*, 2019.

[18] Z. Cheng, J. Li, D. Qi, W. Xiao, H. Junyan, and H. Alexander, "Improving the learning of multi-column convolutional neural network for crowd counting," *ACMMM*, 2019.

[19] H. Cholakkal, G. Sun, F. S. Khan, and L. Shao, "Object counting and instance segmentation with image-level supervision," *CVPR*, 2019.

[20] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *CVPR*, 2017, pp. 1831–1840.

[21] F. Dai, H. Liu, Y. Ma, J. Cao, Q. Zhao, and Y. Zhang, "Dense scale network for crowd counting," *CoRR*, vol. abs/1906.09707, 2019.

[22] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *ICCV*, 2017, pp. 764–773.

[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE Computer Society, 2005, pp. 886–893.

[24] D. Deb and J. Ventura, "An aggregated multicolumn dilated convolution network for perspective-free counting," in *CVPRW*, 2018, pp. 195–204.

[25] X. Ding, Z. Lin, F. He, Y. Wang, and Y. Huang, "A deeply-recursive convolutional network for crowd counting," in *ICASSP*. IEEE, 2018, pp. 1942–1946.

[26] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *TPAMI*, vol. 34, no. 4, pp. 743–761, 2012.

[27] A. Ellis and J. Ferryman, "Pets2010: Dataset and challenge," *AVSS*, pp. 143–150, 2010.

[28] M. Enzweiler and D. M. Gavrila, "Monocular pedestrian detection: Survey and experiments," *TPAMI*, vol. 31, no. 12, pp. 2179–2195, 2009.

[29] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu, "Locality-constrained spatial transformer network for video crowd counting," in *ICME*. IEEE, 2019, pp. 814–819.

[30] H. Farhood, X. He, W. Jia, M. Blumenstein, and H. Li, "Counting people based on linear, weighted, and local random forests," in *DICTA*. IEEE, 2017, pp. 1–7.

[31] L. Fiaschi, U. Köthe, R. Nair, and F. A. Hamprecht, "Learning to count with regression forest and structured labels," in *ICPR*. IEEE, 2012, pp. 2685–2688.

[32] G. French, M. Fisher, M. Mackiewicz, and C. Needle, "Convolutional neural networks for counting fish in fisheries surveillance video," *MVAB*, pp. 1–7, 2015.

[33] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, and C. Zhu, "Fast crowd density estimation with convolutional neural networks," *EAAI*, vol. 43, pp. 81–88, 2015.

[34] C. Gao, P. Li, Y. Zhang, J. Liu, and L. Wang, "People counting based on head detection combining adaboost and cnn in crowded surveillance environment," *Neurocomputing*, vol. 208, pp. 108–116, 2016.

[35] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang, "Cnn-based density estimation and crowd counting: A survey," *arXiv preprint arXiv:2003.12783*, 2020.

[36] J. Gao, Q. Wang, and X. Li, "Pcc net: Perspective crowd counting via spatial convolutional network," *TCSVT*, 2019.

[37] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *TPAMI*, vol. 25, no. 8, pp. 930–943, 2003.

[38] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[39] M. V. Giuffrida, M. Minervini, and S. A. Tsaftaris, "Learning to count leaves in rosette plants," 2016.

[40] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[41] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Onoro-Rubio, "Extremely overlapping vehicle counting," in *PRIA*. Springer, 2015, pp. 423–431.

[42] D. Guo, K. Li, Z.-J. Zha, and M. Wang, "Dadnet: Dilated-attention-deformable convnet for crowd counting," in *ACMMM*. ACM, 2019, pp. 1823–1832.

[43] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *ICLR2016*, 2015.

[44] R. M. Haralick, K. Shanmugam *et al.*, "Textural features for image classification," *TSMC*, no. 6, pp. 610–621, 1973.

[45] G. He, Q. Chen, D. Jiang, X. Lu, and Y. Yuan, "A double-region learning algorithm for counting the number of pedestrians in subway surveillance videos," *EAAI*, vol. 64, pp. 302–314, 2017.

[46] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*. IEEE, 2017, pp. 2961–2969.

[47] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *TPAMI*, vol. 37, no. 9, pp. 1904–1916, 2015.

[48] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, 2015.

[49] M. Hossain, M. Hosseinzadeh, O. Chanda, and Y. Wang, "Crowd counting using scale-aware attention networks," in *WACV*. IEEE, 2019, pp. 1280–1288.

[50] M.-R. Hsieh, Y.-L. Lin, and W. H. Hsu, "Drone-based object counting by spatially regularized regional proposal network," in *ICCV*, 2017, pp. 4145–4153.

[51] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[52] P. Hu and D. Ramanan, "Finding tiny faces," in *Proceedings of the CVPR*. IEEE, 2017, pp. 1522–1530.

[53] Y. Hu, H. Chang, F. Nian, Y. Wang, and T. Li, "Dense crowd counting from still images with convolutional neural networks," *VCIR*, vol. 38, pp. 530–539, 2016.

[54] S. Huang, X. Li, Z. Zhang, F. Wu, S. Gao, R. Ji, and J. Han, "Body structure aware deep crowd counting," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1049–1059, 2017.

[55] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *CVPR*, 2013, pp. 2547–2554.

[56] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *ECCV*, 2018, pp. 532–546.

[57] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025.

[58] S. Jetley, N. A. Lord, N. Lee, and P. H. Torr, "Learn to pay attention," *arXiv*, 2018.

[59] S. Jiang, X. Lu, Y. Lei, and L. Liu, "Mask-aware networks for crowd counting," *TCSVT*, 2019.

[60] X. Jiang, L. Zhang, P. Lv, Y. Guo, R. Zhu, Y. Li, Y. Pang, X. Li, B. Zhou, and M. Xu, "Learning multi-level density maps for crowd counting." *T-NNLS*, 2019.

[61] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. Doermann, and L. Shao, "Crowd counting and density estimation by trellis encoder-decoder networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6133–6142.

[62] A. Josuttes, S. Aich, I. Stavness, C. Pozniak, and S. Shirtliffe, "Utilizing deep learning to predict the number of spikes in wheat (triticum aestivum)," *Phenome 2018 Posters*, vol. 5, p. 8, 2018.

[63] D. Kang and A. Chan, "Crowd counting by adaptively fusing predictions from an image pyramid," *BMVC*, 2018.

[64] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *NIPS*, 2011, pp. 109–117.

[65] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[66] J. Kuen, Z. Wang, and G. Wang, "Recurrent attentional networks for saliency detection," in *CVPR*, 2016, pp. 3668–3677.

[67] S. Kumagai, K. Hotta, and T. Kurita, "Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting,"

*arXiv preprint arXiv:1703.09393*, 2017.

[68] J. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," 2001.

[69] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[70] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *CVPR*, vol. 1. IEEE, 2005, pp. 878–885.

[71] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *NIPS*, 2010, pp. 1324–1332.

[72] H. Li, X. He, H. Wu, S. A. Kasmani, R. Wang, X. Luo, and L. Lin, "Structured inhomogeneous density map learning for crowd counting," *arXiv*, 2018.

[73] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. P. Graf, "Pruning filters for efficient convnets," *ICLR2016*, 2016.

[74] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *ICPR*. IEEE, 2008, pp. 1–4.

[75] S. Z. Li, "Markov random field models in computer vision," in *European conference on computer vision*. Springer, 1994, pp. 361–370.

[76] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *CVPR*, 2018, pp. 1091–1100.

[77] D. Lian, J. Li, J. Zheng, W. Luo, and S. Gao, "Density map regression guided detection network for rgb-d crowd counting and localization," in *CVPR*, 2019, pp. 1821–1830.

[78] M. K. Lim, V. J. Kok, C. C. Loy, and C. S. Chan, "Crowd saliency detection via global similarity structure," in *ICPR*. IEEE, 2014, pp. 3957–3962.

[79] M. Ling and X. Geng, "Indoor crowd counting by mixture of gaussians label distribution learning," *TIP*, vol. 28, no. 11, pp. 5691–5701, 2019.

[80] C. Liu, X. Weng, and Y. Mu, "Recurrent attentive zooming for joint crowd counting and precise localization," in *CVPR*, 2019, pp. 1217–1226.

[81] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann, "Decidenet: Counting varying density crowds through attention guided detection and density estimation," in *CVPR*, 2018, pp. 5197–5206.

[82] L. Liu, S. Amirgholipour, J. Jiang, W. Jia, M. Zeibots, and X. He, "Performance-enhancing network pruning for crowd counting," *Neurocomputing*, vol. 360, pp. 246–253, 2019.

[83] L. Liu, J. Jiang, W. Jia, S. Amirgholipour, M. Zeibots, and X. He, "Denet: A universal network for counting crowd with varying densities and scales," *IEEE Transactions on Multimedia*, 2020.

[84] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, and L. Lin, "Crowd counting with deep structured scale integration network," in *ICCV*, 2019.

[85] L. Liu, H. Wang, G. L. andWanli Ouyang, and L. Lin, "Crowd counting using deep recurrent spatial-aware network," in *IJCAI*, 2018, pp. 849–855.

[86] N. Liu, Y. Long, C. Zou, Q. Niu, L. Pan, and H. Wu, "Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding," in *CVPR*, 2019, pp. 3225–3234.

[87] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer*

*vision*. Springer, 2016, pp. 21–37.

[88] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *CVPR*, 2019, pp. 5099–5108.

[89] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," in *CVPR*, 2018, pp. 7661–7669.

[90] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Exploiting unlabeled data in cnns by self-supervised learning to rank," *TPAMI*, 2019.

[91] Y. Liu, M. Shi, Q. Zhao, and X. Wang, "Point in, box out: Beyond counting persons in crowds," *CVPR*, 2019.

[92] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *ICCV*, vol. 99, no. 2, 1999, pp. 1150–1157.

[93] H. Lu, Z. Cao, Y. Xiao, B. Zhuang, and C. Shen, "Tasselnet: counting maize tassels in the wild via local counts regression network," *Plant methods*, vol. 13, no. 1, p. 79, 2017.

[94] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," *ICCV*, 2017.

[95] J. Luo, J. Wang, H. Xu, and H. Lu, "Real-time people counting for indoor scenes," *Signal Processing*, vol. 124, pp. 27–35, 2016.

[96] J. Ma, Y. Dai, and Y.-P. Tan, "Atrous convolutions spatial pyramid network for crowd counting and density estimation." *Neurocomputing*, vol. 350, pp. 91–101, 2019.

[97] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *ICCV*, 2019, pp. 6142–6151.

[98] A. Makhzani and B. J. Frey, "Winner-take-all autoencoders," in *NIPS*, 2015, pp. 2791–2799.

[99] E. Mamdani, "Application of fuzzy logic to approximate reasoning using linguistic synthesis," in *Proceedings of the sixth international symposium on Multiple-valued logic*. IEEE Computer Society Press, 1976, pp. 196–202.

[100] M. Marsden, K. McGuinness, S. Little, C. E. Keogh, and N. E. O'Connor, "People, penguins and petri dishes: Adapting object counting models to new visual domains and object types without forgetting," in *CVPR*, 2018, pp. 8070–8079.

[101] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," *ICLR2017*, 2017.

[102] M.-h. Oh, P. A. Olsen, and K. N. Ramamurthy, "Crowd counting with decomposed uncertainty," *AAAI*, 2019.

[103] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," in *ECCV*. Springer, 2000, pp. 404–420.

[104] G. Olmschenk, J. Chen, H. Tang, and Z. Zhu, "Dense crowd counting convolutional neural networks with minimal data using semi-supervised dual-goal generative adversarial networks," in *CVPRW*, 2019, pp. 21–28.

[105] G. Olmschenk, H. Tang, and Z. Zhu, "Crowd counting with minimal data using generative adversarial networks for multiple target regression," in *WACV*. IEEE, 2018, pp. 1151–1159.

[106] G. Olmschenk, Z. Zhu, and H. Tang, "Generalizing semi-supervised generative adversarial networks to regression using feature contrasting," *CVIU*, vol. 186, pp. 1–12, 2019.

[107] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *ECCV*. Springer, 2016, pp. 615–629.

[108] N. Paragios and V. Ramesh, "A mrf-based approach for real-time subway monitoring," in *CVPR*, vol. 1. IEEE, 2001, pp. 1–1034.

[109] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," in *ICCV*, 2015, pp. 3253–3261.

[110] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *CVPR*, 2018, pp. 2482–2491.

[111] V. Ranjan, H. Le, and M. Hoai, "Iterative crowd counting," in *ECCV*, 2018, pp. 270–285.

[112] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *CVPR*, 2016, pp. 779–788.

[113] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[114] M. Ren and R. S. Zemel, "End-to-end instance segmentation with recurrent attention," in *CVPR*, 2017, pp. 6656–6664.

[115] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015, pp. 91–99.

[116] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance measures and a data set for multi-target, multi-camera tracking," in *ECCV*. Springer, 2016, pp. 17–35.

[117] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*. Springer, 2015, pp. 234–241.

[118] A. G. Roy, N. Navab, and C. Wachinger, "Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks," in *MICCAI*. Springer, 2018, pp. 421–429.

[119] D. Ryan, S. Denman, C. Fookes, and S. Sridharan, "Crowd counting using multiple local features," in *DITCA*. IEEE, 2009, pp. 81–88.

[120] D. B. Sam and R. V. Babu, "Top-down feedback for crowd counting convolutional neural network," in *AAAI*, 2018.

[121] D. B. Sam, S. V. Peri, A. Kamath, R. V. Babu *et al.*, "Locate, size and count: Accurately resolving people in dense crowds via detection," *arXiv preprint arXiv:1906.07538*, 2019.

[122] D. B. Sam, N. N. Sajjan, H. Maurya, and R. V. Babu, "Almost unsupervised learning for dense crowd counting," in *AAAI*, vol. 27, 2019.

[123] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *CVPR*. IEEE, 2017, pp. 4031–4039.

[124] J. Sang, W. Wu, H. Luo, H. Xiang, Q. Zhang, H. Hu, and X. Xia, "Improved crowd counting method based on scale-adaptive convolutional neural network," *IEEE Access*, 2019.

[125] J. Schlemper, O. Oktay, L. Chen, J. Matthew, C. Knight, B. Kainz, B. Glocker, and D. Rueckert, "Attention-gated networks for improving ultrasound scan plane detection," *arXiv*, 2018.

[126] C. Shang, H. Ai, and B. Bai, "End-to-end crowd counting via joint learning local and global count," in *ICIP*. IEEE, 2016, pp. 1215–1219.

[127] J. Shao, K. Kang, C. Change Loy, and X. Wang, "Deeply learned attributes for crowded scene understanding," in *CVPR*, 2015, pp. 4657–4666.

[128] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *CVPR*, 2018, pp. 5245–5254.

[129] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, and C. Sun, "Crowd counting via weighted vlad on dense attribute feature maps," *TCSVT*, vol. 28, no. 8, pp. 1788–1797, 2018.

[130] M. Shi, Z. Yang, C. Xu, and Q. Chen, "Revisiting perspective information for efficient crowd counting," in *CVPR*, 2019, pp. 7279–7288.

[131] Z. Shi, P. Mettes, and C. G. Snoek, "Counting with focus for free," in *ICCV*, 2019, pp. 4200–4209.

[132] Z. Shi, L. Zhang, Y. Liu, X. Cao, Y. Ye, M.-M. Cheng, and G. Zheng, "Crowd counting with deep negative correlation learning," in *CVPR*, 2018, pp. 5382–5390.

[133] Z. Shi, L. Zhang, Y. Sun, and Y. Ye, "Multiscale multitask deep netvlad for crowd counting," *TII*, vol. 14, no. 11, pp. 4953–4962, 2018.

[134] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[135] V. A. Sindagi and V. M. Patel, "Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *AVSS*. IEEE, 2017, pp. 1–6.

[136] ——, "Generating high-quality crowd density maps using contextual pyramid cnns," in *ICCV*. IEEE, 2017, pp. 1879–1888.

[137] ——, "A survey of recent advances in cnn-based single image crowd counting and density estimation," *Pattern Recognition Letters*, vol. 107, pp. 3–16, 2018.

[138] ——, "Ha-ccn: Hierarchical attention-based crowd counting network," *TIP*, 2019.

[139] ——, "Inverse attention guided deep crowd counting network," *AVSS*, 2019.

[140] ——, "Multi-level bottom-top and top-bottom feature fusion for crowd counting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1002–1012.

[141] V. A. Sindagi, R. Yasarla, and V. M. Patel, "Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method," in *ICCV*, 2019, pp. 1221–1231.

[142] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," *BMVC*, 2015.

[143] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9.

[144] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *CVPR*, 2016, pp. 2818–2826.

[145] Y. Tian, L. Sigal, H. Badino, F. De la Torre, and Y. Liu, "Latent gaussian mixture regression for human pose estimation," in *ACCV*. Springer, 2010, pp. 679–690.

[146] Y. Tian, Y. Lei, J. Zhang, and J. Z. Wang, "Padnet: Pan-density crowd counting," *IEEE Transactions on Image Processing*, 2019.

[147] I. S. Topkaya, H. Erdogan, and F. Porikli, "Counting people by clustering person detector outputs," in *AVSS*. IEEE, 2014, pp. 313–318.

[148] K. Ullrich, E. Meeds, and M. Welling, "Soft weight-sharing for neural network compression," *ICLR2017*, 2017.

[149] V. K. Valloli and K. Mehta, "W-net: Reinforced u-net for density map estimation," *arXiv preprint arXiv:1903.11249*, 2019.

[150] R. R. Varior, B. Shuai, J. Tighe, and D. Modolo, "Scale-aware attention network for crowd counting," *CVPR*, 2019.

[151] R. Viresh, S. Mubarak, and H. N. Minh, "Crowd transformer network," *arXiv preprint arXiv:1904.02774v1*, 2019.

[152] E. Walach and L. Wolf, "Learning to count with cnn boosting," in *ECCV*. Springer, 2016, pp. 660–676.

[153] J. Wan, W. Luo, B. Wu, A. B. Chan, and W. Liu, "Residual regression with semantic prior for crowd counting," in *CVPR*, 2019, pp. 4036–4045.

[154] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *ACM MM*. ACM, 2015, pp. 1299–1302.

[155] Q. Wang, J. Gao, W. Lin, and X. Li, "Nwpu-crowd: A large-scale benchmark for crowd counting," *arXiv:2001.03360v1*, 2020.

[156] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *CVPR*, 2019, pp. 8198–8207.

[157] Q. Wang, J. Wan, and Y. Yuan, "Deep metric learning for crowdedness regression," *TCSVT*, vol. 28, no. 10, pp. 2633–2643, 2018.

[158] Y. Wang and Y. Zou, "Fast visual object counting via example-based density estimation," in *ICIP*. IEEE, 2016, pp. 3653–3657.

[159] Z. Wang, Z. Xiao, K. Xie, Q. Qiu, X. Zhen, and X. Cao, "In defense of single-column networks for crowd counting," *arXiv preprint arXiv:1808.06133*, 2018.

[160] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *ICCV*, 2017, pp. 464–472.

[161] L. Wen, D. Du, P. Zhu, Q. Hu, Q. Wang, L. Bo, and S. Lyu, "Drone-based joint density map estimation, localization and tracking with space-time multi-scale attention network," *arXiv:1912.01811*, 2019.

[162] X. Wu, Y. Zheng, H. Ye, W. Hu, J. Yang, and L. He, "Adaptive scenario discovery for crowd counting," in *ICASSP*. IEEE, 2019, pp. 2382–2386.

[163] F. Xiong, X. Shi, and D.-Y. Yeung, "Spatiotemporal modeling for crowd counting in videos," in *ICCV*, 2017, pp. 5151–5159.

[164] H. Xiong, H. Lu, C. Liu, L. Liang, Z. Cao, and C. Shen, "From open set to closed set: Counting objects by spatial divide-and-conquer," in *ICCV*, 2019.

[165] C. Xu, D. Liang, Y. Xu, W. Zhan, M. Tomizuka, and X. Bai, "Autoscale: Learning to scale for crowd counting," *arXiv:1912.09632*, 2019.

[166] C. Xu, K. Qiu, J. Fu, S. Bai, Y. Xu, and X. Bai, "Learn to scale: Generating multipolar normalized density map for crowd counting," in *ICCV*, 2019.

[167] M. Xu, Z. Ge, X. Jiang, G. Cui, B. Zhou, C. Xu *et al.*, "Depth information guided crowd counting for complex crowd scenes," *PRL*, 2019.

[168] Z. Yan, Y. Yuan, W. Zuo, T. Xiao, Y. Wang, S. Wen, and E. Ding, "Perspective-guided convolution networks for crowd counting," in *ICCV*, 2019.

[169] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *CVPR*, 2016, pp. 5525–5533.

[170] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," *arXiv*, 2016.

[171] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[172] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang, "Multi-scale convolutional neural networks for crowd counting," in *ICIP*. IEEE, 2017, pp. 465–469.

[173] B. Zhan, D. N. Monekosso, P. Remagnino, S. A. Velastin, and L.-Q. Xu, "Crowd analysis: a survey," *MVA*, vol. 19, no. 5-6, pp. 345–357, 2008.

[174] A. Zhang, J. Shen, Z. Xiao, F. Zhu, X. Zhen, X. Cao, and L. Shao, "Relational attention network for crowd counting," in *ICCV*, 2019, pp. 6788–6797.

[175] A. Zhang, L. Yue, J. Shen, F. Zhu, X. Zhen, X. Cao, and L. Shao, "Attentional neural fields for crowd counting," in *ICCV*, 2019, pp. 5714–5713.

[176] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *CVPR*, 2015, pp. 833–841.

[177] H. Zhang, Z. Kyaw, S.-F. Chang, and T.-S. Chua, "Visual translation embedding network for visual relation detection," in *CVPR*, 2017, pp. 5532–5540.

[178] L. Zhang, Z. Shi, M.-M. Cheng, Y. Liu, J.-W. Bian, J. T. Zhou, G. Zheng, and Z. Zeng, "Nonlinear regression via deep negative correlation learning," *TPAMI*, 2019.

[179] L. Zhang, M. Shi, and Q. Chen, "Crowd counting via scale-adaptive convolutional neural network," in *WACV*. IEEE, 2018, pp. 1113–1121.

[180] Q. Zhang and A. B. Chan, "Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns," in *CVPR*, 2019, pp. 8297–8306.

[181] S. Zhang, G. Wu, J. P. Costeira, and J. M. Moura, "Fcn-rlstm: Deep spatio-temporal neural networks for vehicle counting in city cameras," in *ICCV*, 2017, pp. 3667–3676.

[182] ——, "Understanding traffic density from large-scale web camera data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5898–5907.

[183] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *CVPR*, 2016, pp. 589–597.

[184] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890.

[185] M. Zhao, J. Zhang, C. Zhang, and W. Zhang, "Leveraging heterogeneous auxiliary tasks to assist crowd counting," in *CVPR*, 2019, pp. 12 736–12 745.

[186] Z. Zhao, H. Li, R. Zhao, and X. Wang, "Crossing-line crowd counting with two-phase deep neural networks," in *ECCV*. Springer, 2016, pp. 712–726.

[187] B. Zhou, X. Wang, and X. Tang, "Understanding collective crowd behaviors: Learning a mixture model of dynamic pedestrian-agents," in *CVPR*. IEEE, 2012, pp. 2871–2878.

[188] L. Zhu, Z. Zhao, C. Lu, Y. Lin, Y. Peng, and T. Yao, "Dual path multi-scale fusion networks with attention for crowd counting," *arXiv preprint arXiv:1902.01115*, 2019.

[189] P. Zhu, L. Wen, X. Bian, H. Ling, and Q. Hu, "Vision meets drones: A challenge," *arXiv preprint arXiv:1804.07437*, 2018.