

**T.R.**  
**GEBZE TECHNICAL UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**DOMAIN ADAPTATION FOR CROWD COUNTING**

**CENGİZHAN HALDIZ**  
**A THESIS SUBMITTED FOR THE DEGREE OF**  
**MASTER OF SCIENCE**  
**DEPARTMENT OF COMPUTER ENGINEERING**

**GEBZE**  
**2023**

**T.R.**  
**GEBZE TECHNICAL UNIVERSITY**  
**GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**DOMAIN ADAPTATION FOR CROWD  
COUNTING**

**CENGİZHAN HALDIZ**  
**A THESIS SUBMITTED FOR THE DEGREE OF**  
**MASTER OF SCIENCE DEPARTMENT OF**  
**COMPUTER ENGINEERING**

**THESIS SUPERVISOR**  
**PROF. DR. HASARİ ÇELEBİ**

**II. THESIS SUPERVISOR**  
**ASSOC. PROF. DR. ERCHAN APTOULA**

**GEBZE**  
**2023**

**T.C.**  
**GEBZE TEKNİK ÜNİVERSİTESİ**  
**FEN BİLİMLERİ ENSTİTÜSÜ**

**KALABALIK SAYIMI İÇİN ETKİ ALANI**  
**UYARLAMASI**

**CENGİZHAN HALDIZ**  
**YÜKSEK LİSANS TEZİ**  
**BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI**

**DANIŞMANI**  
**PROF. DR. HASARİ ÇELEBİ**

**II. DANIŞMANI**  
**DOÇ. DR. ERCHAN APTOULA**

**GEBZE**  
**2023**

GTÜ Fen Bilimleri Enstitüsü Yönetim Kurulu'nun 19/07/2023 tarih ve 2023/40 sayılı kararıyla oluşturulan jüri tarafından 31/07/2023 tarihinde tez savunma sınavı yapılan Cengizhan HALDIZ'ın tez çalışması Bilgisayar Mühendisliği Anabilim Dalında YÜKSEK LİSANS tezi olarak kabul edilmiştir.

**JÜRİ**

ÜYE

(TEZ DANIŞMANI) : Prof. Dr. Hasari ÇELEBİ

ÜYE (II. DANIŞMANI) : Doç. Dr. Erchan APTOULA

ÜYE : Doç. Dr. Mehmet GÖKTÜRK

ÜYE : Doç. Dr. Mehmet Kemal ÖZDEMİR

ÜYE : Dr. Öğr. Üyesi Burcu YILMAZ

**ONAY**

Gebze Teknik Üniversitesi Fen Bilimleri Enstitüsü Yönetim Kurulu'nun  
...../...../..... tarih ve ...../..... sayılı kararı.

İMZA/MÜHÜR

## SUMMARY

Crowd counting, the task of estimating the number of people in a specific area or scene, has numerous practical applications such as traffic management, public safety, and event planning. Traditional crowd-counting methods rely on manual annotations, which are time-consuming and prone to errors. Recent studies on crowd counting have achieved promising results using Convolutional Neural Network (CNN) architectures. However, due to significant variations in real-world crowd datasets, such as camera angles, resolutions, and crowd densities across different scenes, achieving high performance using standard CNN methods remains challenging. A common issue with such methods is often performance degradation. This problem is known as domain adaptation, and it is an important and active research area in the field of crowd counting with deep learning. To address this challenge, this study proposes a new crowd-counting approach that combines three methods: a guiding network that guides the crowd-counting network and produces normalization parameters across different datasets, the Scale-Adaptive Selection Network which has shown successful results in crowd counting, and the "Distribution Matching" method, which has been proven to be more effective than many Gauss Correction-based methods when used as a loss function. This approach aims to ensure that the trained model performs well across different datasets. Comprehensive experiments on multiple datasets have been conducted, and a performance comparison of the proposed approach against state-of-the-art methods is presented. In addition, experimental studies and comparative results on the use of GAN-based state-of-the-art image-to-image translation methods in the literature for domain adaptation in crowd counting are presented.

**Key Words: Crowd Counting, Density Map, Domain Adaptation, Multi-Level Neural Network, Distribution Matching Count.**

## ÖZET

Belirli bir alandaki veya sahnedeki insan sayısını tahmin etme görevi olan kalabalık sayımı, trafik yönetimi, kamu güvenliği ve etkinlik planlaması gibi çok sayıda pratik uygulamaya sahiptir. Geleneksel kalabalık sayma yöntemleri, zaman alıcı ve hatalara açık olan manuel ek açıklamalara dayanır. Kalabalık sayımı üzerine yapılan son çalışmalar, evrimsel sinir ağı (CNN) mimarilerini kullanarak umut verici sonuçlar elde etmiştir. Ancak, gerçek dünyadaki kalabalık veri kümelerinde kamera bakış açıları, çözünürlükler ve kalabalık yoğunlukları gibi gerçek dünyadaki kalabalık veri kümelerinin farklı sahnelerindeki büyük varyasyonlar nedeniyle, standart CNN yöntemlerini kullanarak yüksek performans elde etmek zor olmaya devam etmektedir. Bu tür yöntemlerin en büyük sorunu genellikle performans düşüşleridir. Bu sorun alan uyarlaması olarak bilinir ve derin öğrenme ile kalabalık sayma alanında önemli ve aktif bir araştırma alanıdır. Bu zorluğun üstesinden gelmek için bu çalışma, üç yöntemi birleştiren yeni bir kalabalık sayma yaklaşımı önermektedir: Kalabalık sayma ağına rehberlik eden ve farklı veri kümelerinde normalleştirme parametrelerini üreten bir kılavuz ağı, kalabalık sayımında başarılı sonuçlar veren Ölçek Uyarlamalı Seçim Ağı ve yitim fonksiyonu olarak kullanıldığında Gauss Düzeltme tabanlı birçok yöntemden daha başarılı olduğu kanıtlanmış "Dağıtım Eşleştirme" yöntemi. Bu yaklaşım, eğitilen modelin farklı veri kümelerinde iyi performans göstermesini sağlamayı amaçlamaktadır. Birden fazla veri kümesi üzerinde kapsamlı deneyler yapılmış ve önerilen yaklaşımın son teknoloji yöntemlerle performans karşılaştırması sunulmuştur. Ayrıca, literatürdeki GAN tabanlı son teknoloji görüntüden görüntüye çeviri yöntemlerinin kalabalık sayımında alan uyarlaması için kullanımına ilişkin deneysel çalışmalar ve karşılaştırmalı sonuçlar sunulmuştur.

**Anahtar Kelimeler:** Kalabalık Sayımı, Yoğunluk Haritası, Etki Alanı Uyarlama, Çok Seviyeli Nöral Ağlar, Dağılım Eşleştirme Sayımı.

## ACKNOWLEDGEMENTS

I would like to express my gratitude to my supervisors, Prof. Dr. Hasari Çelebi and Assoc. Prof. Dr. Erchan Aptoula, for their supports, guidance's, and encouragements throughout this thesis. I am also grateful to my friends and family for their love and support. I would like to note that this work was supported by TUBITAK project (project number: 120N573). Also, the numerical calculations reported in this thesis were partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).



# TABLE of CONTENTS

|                                                                              | <b><u>Page</u></b> |
|------------------------------------------------------------------------------|--------------------|
| SUMMARY                                                                      | v                  |
| ÖZET                                                                         | vi                 |
| ACKNOWLEDMENTS                                                               | vii                |
| TABLE of CONTENTS                                                            | viii               |
| LIST of ABBREVIATIONS and ACRONYMS                                           | x                  |
| LIST of FIGURES                                                              | xi                 |
| LIST of TABLES                                                               | xii                |
| <br>                                                                         |                    |
| 1. INTRODUCTION                                                              | 1                  |
| 1.1. Motivation and Problem Definition                                       | 1                  |
| 1.2. Conducted Research and Contributions                                    | 2                  |
| 1.3. Structure of the Thesis                                                 | 4                  |
| 2. LITERATURE REVIEW                                                         | 5                  |
| 2.1. Introduction                                                            | 5                  |
| 2.2. Datasets                                                                | 8                  |
| 3. CROWD COUNTING VIA MULTI LEVEL NETWORK AND<br>DISTRIBUTION MATCHING COUNT | 11                 |
| 3.1. Introduction                                                            | 11                 |
| 3.2. Convolutional Neural Networks                                           | 12                 |
| 3.3. Detailed Analysis of Scale-Adaptive Selection Network                   | 13                 |
| 3.4. Loss Method                                                             | 16                 |
| 3.5. Training of the Proposed Network                                        | 18                 |
| 3.6. Evaluating Indicator                                                    | 18                 |
| 3.7. Data Preprocessing                                                      | 18                 |
| 3.8. Results                                                                 | 19                 |
| 4. DOMAIN ADAPTATION WITH GUIDING NETWORK                                    | 22                 |
| 4.1. Introduction                                                            | 22                 |
| 4.2. Guiding Network                                                         | 22                 |
| 4.3. Distribution Matching Method                                            | 23                 |
| 4.4. Scale-Adaptive Selection Network                                        | 23                 |

|                                                         |    |
|---------------------------------------------------------|----|
| 4.5. Batch Normalization                                | 24 |
| 4.6. Detailed Analysis of Guiding Network               | 25 |
| 4.7. Training Procedure                                 | 28 |
| 4.8. Experimental Setup                                 | 28 |
| 4.9. Datasets                                           | 29 |
| 4.10. Testing Procedure                                 | 29 |
| 4.11. Results                                           | 30 |
| 5. DOMAIN ADAPTATION WITH IMAGE-TO-IMAGE<br>TRANSLATION | 32 |
| 5.1. Introduction                                       | 32 |
| 5.2. CycleGAN                                           | 33 |
| 5.3. Training Procedure                                 | 34 |
| 5.4. Results                                            | 37 |
| 6. ADAPTIVE KERNEL                                      | 39 |
| 5.1. Introduction                                       | 39 |
| 5.2. Detailed Analysis of Adaptive Kernel Architecture  | 40 |
| 5.3. Testing Procedure                                  | 43 |
| 5.4. Results                                            | 43 |
| 6. CONCLUSIONS AND FUTURE WORKS                         | 45 |
| REFERENCES                                              | 47 |
| BIOGRAPHY                                               | 53 |
| APPENDICES                                              | 54 |

## LIST OF ABBREVIATIONS AND ACRONYMS

| <b><u>Abbreviations</u></b> | <b><u>Explanations</u></b>        |
|-----------------------------|-----------------------------------|
| <b><u>and Acronyms</u></b>  |                                   |
| BN                          | Batch Normalization               |
| CNN                         | Convolutional Neural Networks     |
| DM-Count                    | Distribution Matching Count       |
| GAN                         | Generative Adversarial Networks   |
| GBN                         | Guided Batch Normalization        |
| GPU                         | Graphics Processing Unit          |
| I2I                         | Image to Image                    |
| MAE                         | Mean Absolute Error               |
| OT                          | Optimal Transport                 |
| SASNet                      | Scale-Adaptive Selection Networks |
| TV                          | Total Variation                   |
| RMSE                        | Root Mean Squared Error           |

## LIST OF FIGURES

| <b><u>Figure No:</u></b>                                                             | <b><u>Page</u></b> |
|--------------------------------------------------------------------------------------|--------------------|
| 3.1: General Deep Supervised Method.                                                 | 12                 |
| 3.2: The summary architecture of the SASNet crowd counting model.                    | 14                 |
| 3.3: Density Head.                                                                   | 15                 |
| 3.4: Decoder Block.                                                                  | 15                 |
| 3.5: Qualitative results of the proposed approach on the CityUHK-X dataset.          | 20                 |
| 3.6: Qualitative results of the proposed approach on the FDST dataset.               | 20                 |
| 3.7: Qualitative results of the proposed approach on the VSCrowd dataset.            | 21                 |
| 4.1: General architecture of the proposed crowd counting network.                    | 26                 |
| 4.2: Overview of a GBN block in proposed network.                                    | 27                 |
| 5.1: Image-To-Image Translation Workflow.                                            | 33                 |
| 5.2: General structure of CycleGAN approach.                                         | 34                 |
| 5.3: Sample GCC Image.                                                               | 35                 |
| 5.4: Sample GCC Image After Splitting.                                               | 35                 |
| 5.5: GCC to CityUHK-X Translation Results(CycleGAN).                                 | 36                 |
| 5.6: GCC to CityUHK-X Translation Results(DiscoGAN).                                 | 36                 |
| 5.7: GCC to CityUHK-X Translation Results(S-CycleGAN).                               | 36                 |
| 5.8: GCC to CityUHK-X Translation Results(DualGAN).                                  | 37                 |
| 6.1: Illustration of the weight envelope.                                            | 40                 |
| 6.2: Adaptive Kernel Workflow.                                                       | 42                 |
| 6.3: Use of Adaptive Convolution in the decoding block on the proposed architecture. | 43                 |
| B.1: VGG-16 Network Architecture.                                                    | 55                 |
| B.2: VGG Block.                                                                      | 56                 |

# LIST OF TABLES

| <b><u>Table No:</u></b>                                                | <b><u>Page</u></b> |
|------------------------------------------------------------------------|--------------------|
| 2.1: Crowd Counting Datasets.                                          | 9                  |
| 3.1: Counting Performance on CityUHK-X, VSCrowd, and<br>FDST Datasets. | 19                 |
| 4.1: Domain Adaptation Result over VSCrowd Dataset.                    | 30                 |
| 4.2: Domain Adaptation Result over FDST Dataset.                       | 30                 |
| 4.3: Domain Adaptation Result over CityUHK-X Dataset.                  | 31                 |
| 5.1: Image-to-Image Translation Datasets.                              | 34                 |
| 5.2: Domain Adaptation Results for GCC to CityUHK-X<br>Scenario.       | 37                 |
| 6.1: Adaptive Kernel Results.                                          | 44                 |

# 1. INTRODUCTION

## 1.1. Motivation and Problem Definition

In sociology, psychology and safety, the analysis of crowd dynamics is of interest. Computer vision can perform many interesting analyses [66], for example, to learn the evolution of crowd flow and floor fields [1], track an individual within a group [46], segment a group into semantic regions [37], [71], detect salient areas in a group [36], or recognize anomalous patterns in crowds [23], [39]. One of the most important tasks in crowd analysis, which has a wide range of applications, is to count people and track their behavior over time.

Public safety and security is one of the main application areas for crowd counting. Large crowds are often involved in tragedies, particularly during religious, political and musical events. A 2010 crowd crush at Germany's Love Parade music festival resulted in the death of 21 people and injured many others. A stampede occurred near the Sabarimala Temple in India, with a death toll of over 100. The tragedies that have occurred could be prevented if the site was designed in a way to make it safer and crowd control was more effective. Video-based crowd counting is a powerful tool to detect overcrowding situations and facilitate better crowd control. This tool also assists in profiling population movements over time and between spaces to establish global situational awareness. It can be used for developing long-term strategies of crowd management and designing evacuation routes.

Crowd counting is a useful intelligence tool in retail [54] that can provide a valuable indication of customer interest by quantifying the number and length of people browsing products, or the percentages of visitors to the store at different times during the day. This information can be used to optimize the floor plan, staffing, and product displays.

The traditional crowd-counting methods rely on manual annotation, which is time-consuming and prone to errors. In recent years, convolutional neural network architecture-based crowd-counting approaches have achieved impressive results [2] and they can get promising results. One of the challenges in crowd counting with deep learning is the difficulty of adapting models trained on large variations in the different

scenes of the real-world crowd datasets, such as camera viewpoints, resolutions, and crowd densities [14].

To tackle the aforementioned issues many deep-learning approaches have been applied. One of these methods is Distribution-Matching-Count (DM-Count) which has been proposed in [55], this method is dependent on applying new loss functions based on optimal transport to diminish the distribution discrepancy by measuring the similarity between the density map of the predicted and the ground truth. Moreover, the standard supervised crowd counting model which is called multi-level-based Scale Adaptive Selection Network (SASNet) was presented by [43] in order to fully use multi-scale feature representations. Another supervised crowd-counting model CSRNet has been proposed by [29], which depends on the combination of standard convolutional neural networks (CNN) and dilated CNNs which are used to remove the pooling operation. Additionally, Reddy et al. [45] introduced guiding networks as a novel approach to address the issue of scene shift between two disparate datasets. Guiding networks utilize models that can adapt to various scene properties and have been shown to enhance the performance of crowd-counting algorithms in real-world scenarios. The network consists of two parts: the main crowd-counting network and the guiding network. The guiding network guides the main crowd-counting network through the use of multi-batch normalization layers, adjusting the normalization parameters during the training phase.

While the aforementioned methods work well with specific datasets and fail to generalize to datasets with large variations, this thesis work proposes a new crowd counting method inspired by three pioneering works, including GuidedBN [45], SASNet [43], and DM-Count [55], to overcome this difficulty. The proposed architecture is based on GuidedBN, substituting the main crowd counting module with SASNet, and uses DM-Count loss functions instead of the standard loss function for model training.

## **1.2. Conducted Research and Contributions**

Recent studies on crowd counting have demonstrated promising results using Convolutional Neural Networks (CNNs). Unfortunately, achieving high performance with standard CNN methods remains challenging due to large variations between real-

world crowd datasets - camera angles, resolutions and densities can vary considerably, as well as performance degradation due to these methods.

To overcome these hurdles, this thesis proposes an innovative crowd counting approach that integrates three methods. A Guiding Network [45] guides and generates normalization parameters across datasets while scaling adaptive selection networks have demonstrated promising results when applied to crowd counting; Scale-Adaptive Selection Networks [43] have proven themselves adept at crowd counting; while Distribution Matching [55] as loss function has proven more successful than many other techniques.

This thesis's primary objective is to develop a crowd counting model capable of accurately estimating the number of people present in different scenes regardless of variations in crowd densities, camera perspectives or resolutions. Specific objectives include:

- To use a Guiding Network that can guide the crowd counting network and support domain adaptation by creating normalization parameters on different datasets.
- To implement a Scale-Adaptive Selection Network that can adapt to the varying scales of the crowd in different images.
- To use a Distribution Matching method as a loss function to optimize the performance of the crowd counting model.
- To evaluate the performance of the proposed crowd counting model on multiple datasets and compare it with state-of-the-art methods.
- To explore the potential of synthetic datasets for domain adaptation, emulating real-world datasets through Image-to-Image translation techniques.
- To evaluate the effect on the accuracy of crowd estimation of the network determining the optimal core size based on the training dataset, rather than sticking to a fixed core size, within the crowd counting architecture.

### 1.3. Structure of the Thesis

This thesis is structured into 7 chapters.

Chapter 2 provides an extensive literature review on the topic of crowd analysis. This chapter also details the datasets that are currently available for this field of study.

In Chapter 3, a new network architecture is proposed using the SASNet crowd-counting network proposed by [43] as a multi-level network and the DM-Count method inspired by the work in [55] as a loss function.

In Section 4, inspired by the Guiding Network used in [45] as a field adaptation method, it is added to the crowd-counting network architecture proposed in Section 3 and the results of the experiments are shared.

In Chapter 5, studies on domain adaptation to real-world datasets using synthetic datasets and state-of-the-art image-to-image translation methods and results obtained using different methods in the literature are presented.

In Section 6, some of the decoding layers in SASNet, the crowd-counting network proposed in Section 3, are replaced with convolution layers with the adaptive kernel feature proposed in [53] to enable the network to learn the optimal kernel sizes for the trained data set.

Chapter 7 concludes the thesis and offers suggestions for potential future research directions. Additionally, there are appendices included which detail the internal structure of the VGG-16 network architecture.

## 2. LITERATURE REVIEW

### 2.1. Introduction

Crowd counting, the task of estimating the number of individuals in a scene, has become a significant research area due to its importance in various applications such as public safety, surveillance, and event management. With the advent of deep learning techniques, the performance of crowd counting models has seen significant improvements. However, the challenge of domain adaptation in crowd counting remains a critical issue.

In this section, the studies found in the literature concerning crowd counting and domain adaptation in crowd counting have been reviewed.

Crowd density estimation is a highly intensively researched topic [51], with a plethora of existing algorithms. These algorithms are generally classified as detection-based, regression-based, and convolutional neural network (CNN)-based. The first two rely on handcrafted low-level features (e.g. local binary patterns (LBP), and histogram of oriented gradients (HOG), etc.). Detection-based methods in particular strive to count pedestrians through their entire or partial (head, shoulder) body appearance [9], [12]. Consequently, they have limited recognition performance as they suffer from severe occlusions and only perform satisfactorily with sparse crowds. To address the limitations of detection-based methods, regression-based methods have been introduced. The main idea consists of mapping low level features extracted from local image patches to the sought count or more often to crowd density. A wide variety of regression methods has been investigated in the literature, such as linear regression [41], piece-wise linear regression, ridge regression [6], etc. Both of these traditional approaches however are limited by the representation capacity of low-level image features.

CNNs on the other hand, possess a powerful ability for learning visual data representations. Ever since the pioneering work of [7], a huge variety of network architectures has been developed to address this issue. The main drawback of CNN based approaches is that they require in some form the perspective information which is not always available. Hence, the introduced methods often aim to remedy this inconvenience. Notably, [69] have proposed a multi-column CNN wherein various

receptive fields of different sizes are merged together at the last layer, thus performing multi-scale recognition. On the other hand [48], have proposed the Switch-CNN, consisting of multiple independent CNN regressors with distinct receptive fields, where the switch classifier selects the optimal regressor for each image patch. More recently, [44] proposed an iterative crowd counting framework. This framework first generates a low-quality density map, which gradually evolves to a high-quality density map. In [29], a novel network known as CSRNet is presented, specifically designed for the task of Congested Scene Recognition. This innovative network leverages a combination of data-driven techniques and deep learning methodologies to interpret densely populated scenes, provide precise count estimations, and construct detailed density maps. The proposed CSRNet's architecture is divided into two main parts: a front-end Convolutional Neural Network (CNN) designed for 2D feature extraction, and a back-end dilated CNN. The back-end dilated CNN employs dilated kernels to increase the receptive fields, serving as a substitute for conventional pooling operations. Moreover, [64] introduced a PanDensity Network consisting of several components, targeting increased local estimation accuracy. The work by [60] achieved outstanding results via a spatial divide-and-conquer strategy, thus forcing image patches to always contain counts present at training. The research in [34] developed an end-to-end trainable deep architecture for crowd counting. This architecture amalgamates features obtained using multiple receptive field sizes and learns the significance of each feature at each image location. The study in [47] proposes a detection framework for dense crowd counting. This framework employs an architecture that identifies everyone in the crowd, sizes the detected heads with a bounding box, and then counts them. This approach negates the need for the commonly used density regression paradigm.

The crowd density estimation is a challenging problem due to the following reasons:

- Feeding from images at different resolutions, perspective angles, scales, lighting, weather changes.
- The crowd densities can vary greatly.
- When the video is fed from images, it must be capable of working in real time.
- Generalization to crowd counts beyond those seen at training is required.

- It should produce a solution against the domain shifting effect between different datasets.

The majority of existing methods for crowd counting are designed to work in a particular domain. It is therefore a difficult but important problem to design crowd counting models capable of high performance in any domain. There is some robust crowd counting approaches against domain shifts proposed in recent years [17], [24], [32], [62], [70], [68].

CVCS [68] proposes an arbitrary camera location cross-view multi-view crowd-counting paradigm where training and test sets are taken from different scenes. CVCS can select and fuse multiple views by using the camera layout geometry and noise view regularization to deal with non-correspondence. CDCC [36] proposes an neural linear transformation method that uses biases and domain factors to learn domain shift. AdaCrowd [45] uses a crowd-counting network and a guidance network that predicts certain parameters in the counting networks based on unlabeled data.

The study referenced in [14] presents a crowd counting technique that employs a domain-adaptation approach, utilizing multilevel feature-aware adaptation and a structured density map alignment module. This method is trained on data generated with ground-truth specific to real-world scenarios. In contrast, the research in [13] suggests learning from synthetic data and transferring this knowledge to real data, even in the absence of ground truth. The DACC framework, as part of this research, employs high-quality image translation and density map reconstruction to enhance the quality of cross-domain crowd count. Lastly, the research in [3] introduces a two-step approach that leverages intra-domain knowledge to support unsupervised, cross-domain crowd counting using synthetic datasets.

Scale or density gaps between datasets are another type of domain-gap for domain adaptive counting [15], [38], [59], [61]. Scale alignment modules are used in the work of [38] to propose a crowd counting model universal that can be used across scenes and datasets. DCANet [61] is a domain-guiding network of channel attention that guides the extraction and representation of domain-specific features for multi domain counting. DKPNet [5] is a domain-specific network that propagates knowledge to improve crowd counting.

## 2.2. Datasets

Some datasets found in the literature for crowd counting tasks are shown in Table 2.1. The table shows the year of publication of the dataset, its intended task, its category, some notes about the dataset and publishing papers or documents of the datasets are given in the Table.



Table 2.1: Crowd Counting Datasets.

|   | <b>Dataset</b>            | <b>Year</b> | <b>Task</b>      | <b>Type</b> | <b>Notes</b>                                                                                                                                         |
|---|---------------------------|-------------|------------------|-------------|------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | FDST(Fudan- ShanghaiTech) | 2019        | Crowd Statistics | Video       | The dataset includes a total of 394K labeled pedestrians, including 60 videos for training and 40 videos for testing. 1920x1080 resolution. [11]     |
| 2 | Mall                      | 2013        | Crowd Statistics | Video       | The PartA dataset has 483 images and the PartB dataset has 716 images, for a total of 330,165 labeled springs.[35]                                   |
| 3 | ShangaiTech               | 2016        | Crowd Statistics | Image       | PartA: 482 images. PartB: 716 images. 330,165 annotations in total. [69]                                                                             |
| 4 | VSCrowd                   | 2022        | Crowd Statistics | Video       | PartA dataset contains a total of 482 images and PartB dataset contains a total of 716 images. In total, there are 330,165 labeled pedestrians. [28] |
| 5 | WorldExpo'10              | 2015        | Crowd Statistics | Video       | 3980 images with labels, covering 108 different camera shots. Train : 3380 images over 103 shots. Test : 600 images over 5 scenes. [7]               |

Table 2.1: Continued.

|    |            |      |                  |           |                                                                                                                                                |
|----|------------|------|------------------|-----------|------------------------------------------------------------------------------------------------------------------------------------------------|
| 6  | CityUHK-X  | 2017 | Crowd Statistics | Video     | The dataset includes a total of 106K labeled pedestrians, including 43 scenes for training and 12 scenes for testing. 384×512 resolution. [21] |
| 7  | DroneCrowd | 2019 | Crowd Statistics | Video     | A total of 33,600 images and 4,864,280 labeled pedestrians. 1080x1920 resolution. [57]                                                         |
| 8  | CityStreet | 2019 | Crowd Statistics | Video     | A total of 500 images. 1520x2704 resolution. [67]                                                                                              |
| 9  | UCF-CC-50  | 2013 | Crowd Statistics | Image     | A total of 50 images and 63,974 labeled pedestrians. 2101x2888 resolution. [18]                                                                |
| 10 | UCSD       | 2008 | Crowd Statistics | Video     | A total of 2000 images and 49,885 labeled pedestrians. 158x238 resolution. [4]                                                                 |
| 11 | GCC        | 2019 | Crowd Statistics | Synthetic | 15,212 synthetic images with labels, covering 400 different scenes. 1920x1080 resolution. [56]                                                 |

# 3. CROWD COUNTING VIA MULTI LEVEL NETWORK AND DISTRIBUTION MATCHING COUNT

## 3.1. Introduction

Crowd counting, a critical component in various domains such as surveillance, public safety, and event management, has been a challenging task due to factors such as occlusion, perspective distortion, and varying crowd densities. Recent advancements in deep learning have significantly improved the accuracy and efficiency of crowd counting techniques. This chapter aims to further enhance these techniques by integrating two state-of-the-art methods: Scale-Adaptive Selection Network (SASNet) [43] and Distribution Matching(DM-Count) [55].

SASNet, is a novel crowd counting approach that addresses the issue of scale variation, a common challenge in crowd counting. This method utilizes a scale aggregation module to fuse multi-scale features effectively. SASNet has demonstrated superior performance in handling scale variations and has achieved state-of-the-art results on several benchmark datasets.

While SASNet provides an effective way to handle scale variations, the choice of loss function also plays a crucial role in the performance of a crowd counting model. Traditional crowd counting methods often use a Gaussian to smooth each annotated dot in the training images, which can hurt the model's generalization performance. To address this, incorporated DM-Count, a method that uses Distribution Matching for crowd counting. DM-Count employs Optimal Transport (OT) to measure the similarity between the normalized predicted density map and the normalized ground truth density map, providing a tighter generalization error bound than Gaussian smoothed methods. Furthermore, DM-Count has shown to outperform previous state-of-the-art methods on several large-scale counting datasets.

By integrating SASNet and DM-Count, this chapter aims to develop a more robust and accurate crowd counting model. The proposed model is expected to handle scale variations effectively and provide a tighter generalization error bound, thereby improving the overall performance of crowd counting.

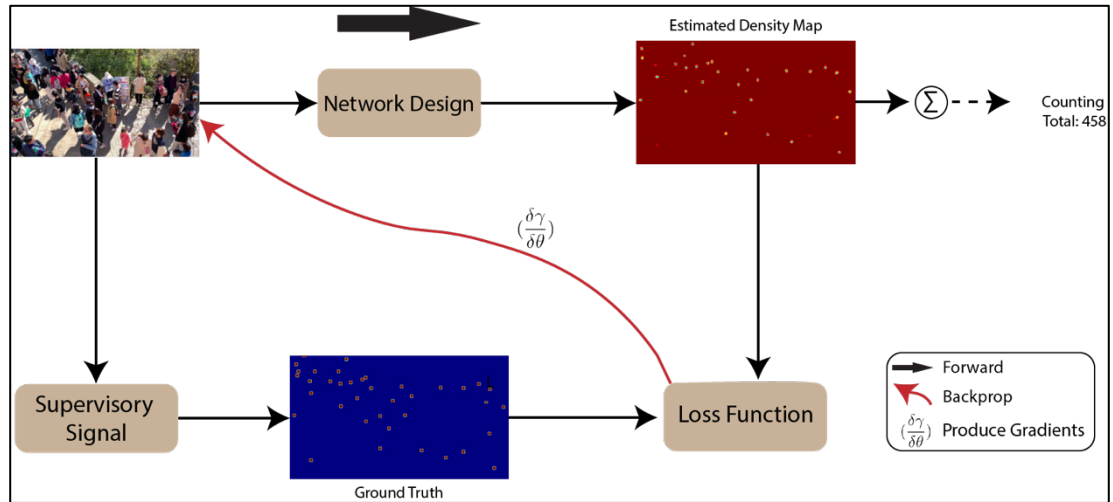


Figure 3.1: General Deep Supervised Method.

## 3.2. Convolutional Neural Networks

LeCun et al.'s (1998) introduction of Convolutional Neural Networks (CNNs), now widely utilized within deep learning, especially for image recognition and analysis tasks [27], has led them to become an indispensable element. Convolutional Layers form the core component of these networks, offering advanced algorithms capable of learning spatial hierarchies from data input. Convolutional layers work by sliding a filter or kernel over input data, applying convolution operations at each step [26]. This operation involves element-wise multiplication between filter values and input values, followed by their summation, yielding a feature or convolutional map. Convolutional Layers use filters trained to recognize specific features in input data, such as colors, edges or textures. Their output - known as a feature map - contains higher values where such features have been detected [65]. The Convolutional Layer employs two key techniques, stride and padding. Stride refers to the step size of a kernel; padding involves adding extra pixels around input data so as to ensure a suitable kernel fits [10]. Selecting appropriate stride and padding values is critical to the network's performance and feature map quality [16]. Convolutional Layer's strength lies in its ability to extract high-level features from input data. As networks deepen, these layers become essential tools for object detection and image segmentation [25]. Furthermore, Convolutional Layers help reduce the dimensionality of input data sets [50]. CNNs can save computational resources while helping reduce overfitting by employing pooling layers in their approach [50]. By doing this, CNNs

can reduce spatial dimensions (width and height) while maintaining key features [52]. Taking this approach also helps avoid overfitting [20], [40]. CNNs provide another benefit when used with Convolutional Layers; these layers can manage translation invariance by recognizing features regardless of their position in the input data [10,11,12,13,14,15,17,18,19,10]. This attribute highlights their efficiency for image data processing. Convolutional Layers provide the foundation of Convolutional Neural Networks [40] Rectified Linear Units Improve Restricted Boltzmann Machines [30]. By combining with other layers such as Rectified Linear Units (ReLUs) to introduce nonlinearity and fully connected layers for classification tasks, Convolutional Neural Networks become extremely efficient machines [30], [49]. Convolutional Layers play an essential part in deep learning due to their capacity for discovering, recognizing and mapping intricate patterns and features from input data. Convolutional Layers have many applications ranging from facial recognition and autonomous driving to medical image analysis [30], [49].

### **3.3. Detailed Analysis of Scale-Adaptive Selection Network**

The Scale-Adaptive Selection Network (SASNet) [43] is an innovative approach that provides an effective solution to large scale variations in crowd counting problems. SASNet leverages multi-scale feature representations in a multi-level network to address the issue of scale variation, which is a significant challenge in crowd counting. The SASNet consists of two primary components: a Density Branch and a Confidence Branch. The Density Branch comprises five density heads, each of which makes predictions using features from different levels. Each density head collects multi-scale context information using a multi-branch module, which consists of three convolution branches with different convolution kernel sizes and a skip connection branch. The output features are concatenated along the channel dimension and then a 1x1 convolution is used to obtain the prediction for the  $i$ -th feature level. The Confidence Branch, on the other hand, contains a confidence head for each feature level. Each confidence head estimates a score that represents its confidence in being the most appropriate feature level for a specific image patch. The confidence maps are generated using the features from the corresponding feature levels. One of the key strategies of SASNet is its selection strategy. After obtaining the output density maps,

the strategy fully leverages multi-scale feature representations to select the estimation with the minimum counting error for a specific patch. However, during inference, it's not possible to determine the most appropriate feature level with the minimum counting error since there's no access to the ground truth. Therefore, in this approach, the Confidence Branch has been used to determine the most appropriate feature level for a specific patch.

In conclusion, the SASNet presents a scale-adaptive approach for crowd counting. This approach aims to improve the accuracy and precision of crowd counting by considering predictions from various feature levels and selecting the most appropriate feature level for a specific patch. Furthermore, this approach helps to mitigate the challenges brought by scale variation, especially in the counting of large-scale crowds.

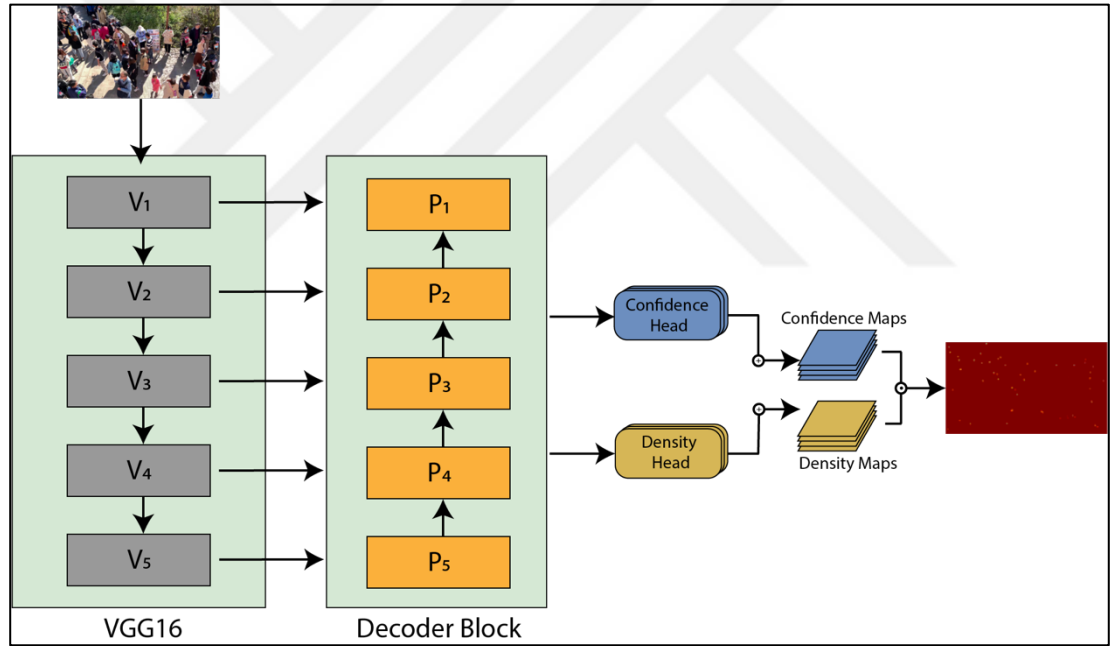


Figure 3.2: The summary architecture of the SASNet crowd counting model.

The SASNet architecture is primarily composed of three components: the U-shape branch, the confidence branch, and the density branch. First, the U shape backbone is used to extract multi-level representations of features for an image. These features are then fed into confidence heads and densities heads to produce multi-level density maps and confidence maps. Using the multilevel confidence maps as a guide combines multilevel density maps using a weighted average to get the final prediction. The summary architecture of the network is presented in Figure 3.2. In addition, the

structure of the network for the density head and decoder block are presented in Figure 3.3 and Figure 3.4 respectively. Conv $n$  is a convolution of kernel size  $n \times n$ .

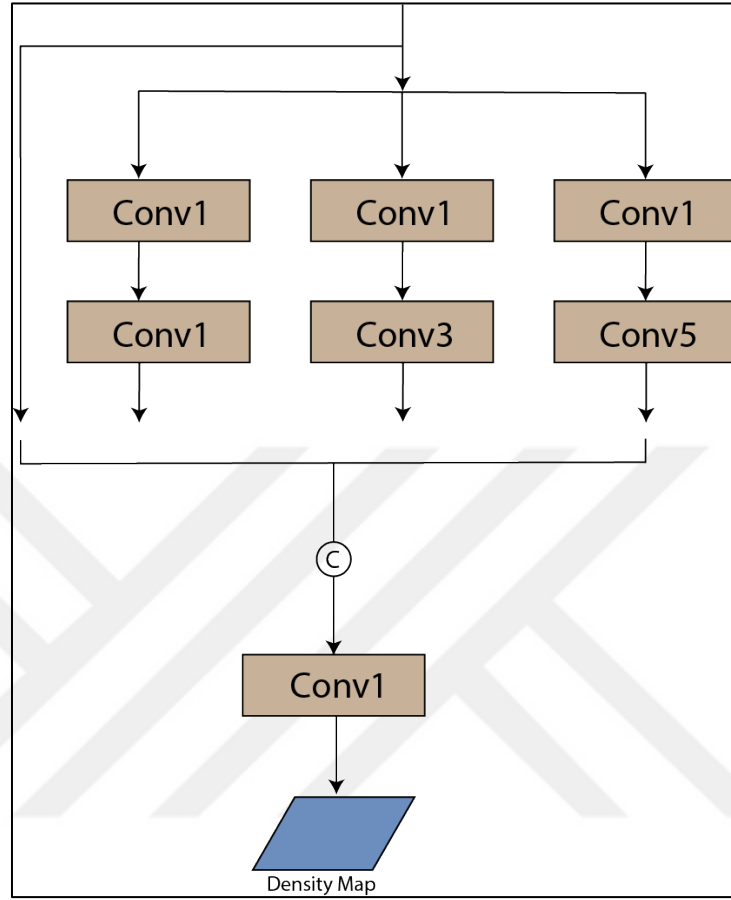


Figure 3.3: Density Head.

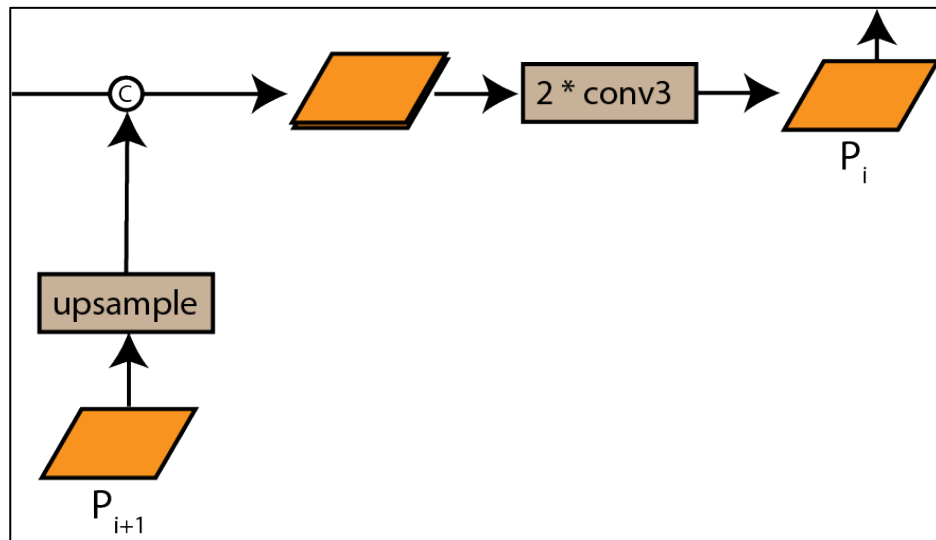


Figure 3.4: Decoder Block.

### 3.4. Loss Method

In the domain of image processing for crowd estimation, an image serves as input data to an algorithmic model, subsequently generating a density map as output. The aggregate count is derived algorithmically by summing values across this computed density map. In the current computational approach, leveraging principles from the DM-Count method [55]—which introduces an innovative loss function to assess the congruence between computed and benchmark density maps—the task of crowd estimation is modeled as a distribution alignment problem. The loss function is constructed algorithmically using three components: the Optimal Transport (OT) loss, a count discrepancy loss, and the Total Variation (TV) regularization. Here, Optimal Transport is a computational measure of the cost-efficiency to morph one probability distribution into another. For the OT loss computation, both the resultant density map from the model's output layer and the benchmark density map are transformed into probability distribution formats. The distance between these distributions, in a computational space, is measured using the Sinkhorn distance approach [8]. The OT loss is algorithmically defined as presented in equation (3.1).

$$l_{OT}(z, \hat{z}) = \mathcal{W}\left(\frac{z}{|z|_1}, \frac{\hat{z}}{|\hat{z}|_1}\right) \quad (3.1)$$

Here,  $l_{OT}$  represents the optimal transport cost,  $\mathcal{W}$  signifies the Wasserstein distance,  $\hat{z}$  is the estimated density map output from the final layer of a neural network, and  $z$  is the vectorized discrete map of actual reference values. The optimal transport cost can be computed as follows using the Sinkhorn distance algorithm.

$$l_{OT}(z, \hat{z}) = \mathcal{W}\left(\frac{z}{|z|_1}, \frac{\hat{z}}{|\hat{z}|_1}\right) = \left\langle \alpha^*, \frac{z}{|z|_1} \right\rangle + \left\langle \beta^*, \frac{\hat{z}}{|\hat{z}|_1} \right\rangle \quad (3.2)$$

In this context,  $\alpha$  and  $\beta$  are the resultant values from the Sinkhorn algorithm execution. Following this, the gradient for back-propagation within the neural network architecture is determined.

$$\frac{\partial l_{OT}(z, \hat{z})}{\partial \hat{z}} = \frac{\beta}{|\hat{z}|} - \frac{\langle \beta, \hat{z} \rangle}{|\hat{z}|^2} \quad (3.3)$$

Moreover, the loss function incorporates the widely-used count loss (3.4), as well as the Total Variation (TV) loss (3.5) typically employed in Generative Adversarial Networks.

$$l_c(z, \hat{z}) = ||z|_1 - |\hat{z}|_1| \quad (3.4)$$

$$l_{TV}(z, \hat{z}) = \left| \frac{z}{|z|_1} - \frac{\hat{z}}{|\hat{z}|_1} \right|_{TV} = \frac{1}{2} \left| \frac{z}{|z|_1} - \frac{\hat{z}}{|\hat{z}|_1} \right|_1 \quad (3.5)$$

Here,  $|\cdot|_1$  represents the  $L_1$  norm of a vector. In this case,  $|z|_1$  and  $|\hat{z}|_1$  denote the actual and estimated counts, respectively.

In the formulation presented in equation (3.6), the composite loss function is articulated through three distinct components: the count loss (LC), representing the absolute discrepancy between the ground truth density map  $z$  and the predicted output  $\hat{z}$  from the neural architecture; the Optimal Transport (OT) loss, a metric quantifying the congruence between two probability distributions via the Wasserstein distance; and the Total Variation (TV) loss, a regularization technique employed to bolster the robustness of the training iteration.

$$l(z, \hat{z}) = \lambda_1 l_c(z, \hat{z}) + \lambda_2 l_{OT}(z, \hat{z}) + \lambda_3 |z|_1 l_{TV}(z, \hat{z}) \quad (3.6)$$

Here,  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are adjustable hyperparameters for the LC, OT, and TV losses respectively. Through empirical evaluation, these hyperparameters have been configured to the following values:  $\lambda_1 = 1$ ,  $\lambda_2 = 0.1$ ,  $\lambda_3 = 0.01$ .

### 3.5. Training of the Proposed Network

The model undergoes training in an end-to-end manner, leveraging the loss function delineated in the preceding section. The weights of the backbone are instantiated from pre-trained VGG-16 weights.

### 3.6. Evaluating Indicator

In alignment with preceding studies in the realm of crowd counting, the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) metrics are employed to gauge the efficacy of comparative methodologies in the experiments presented. The definitions of these metrics are encapsulated within the subsequent equations.

$$MAE = \frac{1}{S} \sum_{i=1}^S |C_i - \hat{C}_i| \quad (3.7)$$

$$RMSE = \sqrt{\frac{1}{S} \sum_{i=1}^S (C_i - \hat{C}_i)^2} \quad (3.8)$$

Within this context,  $S$  represents the aggregate number of test samples.  $C_i$  and  $\hat{C}_i$  denote the ground truth and the estimated counts of individuals for the  $i$ -th sample, respectively, ascertained by aggregating over the entirety of the density map. Broadly speaking, while MAE serves as an indicator of the precision in crowd counting, RMSE offers insights into the resilience of the estimation in the face of outliers.

### 3.7. Data Preprocessing

In this step, images obtained from various sources are standardized. In addition, random cropping was applied for data augmentation. For this purpose, both the images and the ground-truth density maps were cropped to sizes of  $540 \times 540$  for VSCrowd and FDST, and  $216 \times 216$  for CityUHK-X datasets.

### 3.8. Results

In the ensuing subsection, the method delineated above is juxtaposed with contemporary approaches. It is noteworthy that the training phase is executed on the CityUHK-X, FDST, and VSCrowd training datasets.

Table 3.1: Counting Performance on CityUHK-X, VSCrowd, and FDST Datasets.

| Model         | CityUHK-X [21] |             | VSCrowd [28] |             | FDST [11]   |             |
|---------------|----------------|-------------|--------------|-------------|-------------|-------------|
|               | RMSE           | MAE         | RMSE         | MAE         | RMSE        | MAE         |
| CSRNet [29]   | 16.4           | 10.9        | 21.1         | 13.8        | 5.27        | 4.04        |
| Adacrowd [45] | 16.1           | 10.8        | 47.1         | 26.06       | 3.1         | 2.4         |
| RAZNet [31]   | -              | -           | 19.5         | 12.7        | 5.5         | 4.7         |
| S-GNANe [28]  | -              | -           | <b>12.3</b>  | 9.9         | 2.9         | 2.1         |
| STANet [58]   | -              | -           | 18.6         | 12.0        | 5.1         | 3.6         |
| EPF [33]      | 15.38          | 11.07       | 14.6         | 10.4        | 2.6         | 2.2         |
| LST [11]      | -              | -           | 41.1         | 22.9        | 4.5         | 3.4         |
| DM-Count [55] | 27.39          | 19.44       | 18.12        | 8.85        | <b>2.45</b> | <b>1.81</b> |
| SFCN [56]     | 13.8           | 9.7         | 20.1         | 9.3         | 2.5         | 1.9         |
| Our           | <b>12.55</b>   | <b>8.24</b> | 15.58        | <b>8.84</b> | 2.5         | 1.9         |

Table 3.1 delineates the mean outcomes derived from comprehensive scene training and evaluation on the CityUHK-X [21], VSCrowd [28], and FDST [11] datasets. In the context of CityUHK-X, the MAE and RMSE values achieved by the introduced method marginally surpass those of the baseline methodologies. For VSCrowd, it is observed that the MAE of the introduced method stands superior among all contemporary techniques, though it does not exceed the performance of the EPF [33] method. Concerning the FDST [11] dataset, the introduced methodology does not surpass DM-Count [55]. It is pertinent to mention that the symbol '-' is utilized in scenarios where the source code remains inaccessible.

In addition, visual results of some examples of CityUHK-X, FDST, and VSCrowd datasets are shared below.

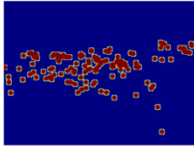
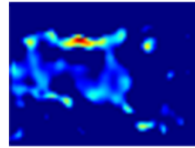
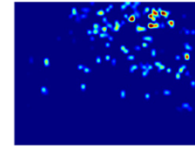
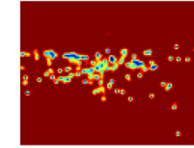
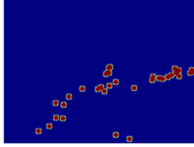
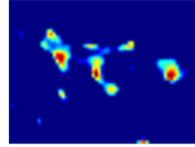
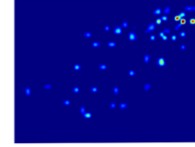
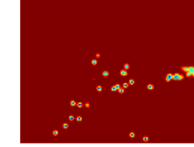
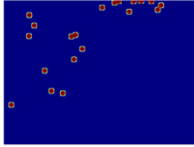
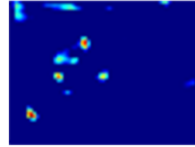
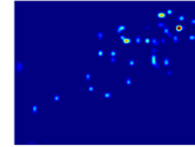
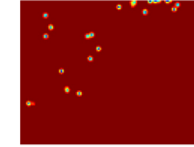
| Sample                                                                            | Ground Truth                                                                      | DM-Count                                                                          | EPF                                                                                | Our Approach                                                                        |
|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|  |  |  |  |  |
|                                                                                   | GT: 118                                                                           | Prediction: 70.59                                                                 | Prediction : 78.68                                                                 | Prediction: 111.43                                                                  |
|  |  |  |  |  |
|                                                                                   | GT: 34                                                                            | Prediction: 28                                                                    | Prediction: 46.17                                                                  | Prediction: 35.17                                                                   |
|  |  |  |  |  |
|                                                                                   | GT: 20                                                                            | Prediction: 13.55                                                                 | Prediction: 22.58                                                                  | Prediction: 19.6                                                                    |

Figure 3.5: Qualitative results of the proposed approach on the CityUHK-X dataset.

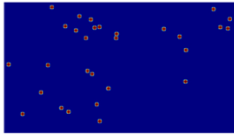
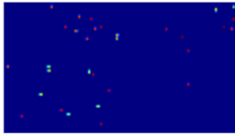
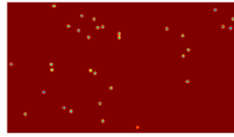
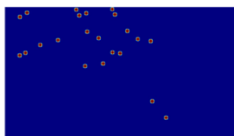
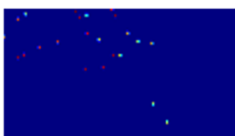
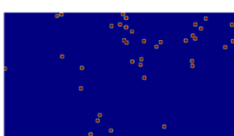
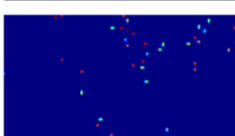
| Sample                                                                              | Ground Truth                                                                        | DM-Count                                                                             | Our Approach                                                                          |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------|
|  |  |  |  |
|                                                                                     | GT: 29                                                                              | Prediction: 30.7                                                                     | Prediction : 31.42                                                                    |
|  |  |  |  |
|                                                                                     | GT: 22                                                                              | Prediction: 23.17                                                                    | Prediction : 21.02                                                                    |
|  |  |  |  |
|                                                                                     | GT: 37                                                                              | Prediction: 32.27                                                                    | Prediction : 37.03                                                                    |

Figure 3.6: Qualitative results of the proposed approach on the FDST dataset.

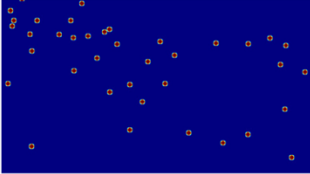
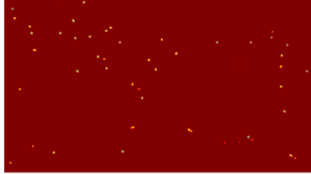
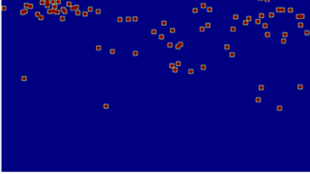
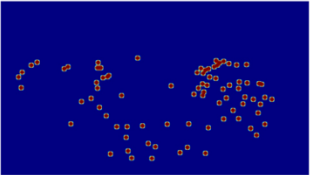
| Sample                                                                            | Ground Truth                                                                       | Our Approach                                                                        |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------|
|  |  |  |
|                                                                                   | GT : 29                                                                            | Prediction : 31.42                                                                  |
|  |  |  |
|                                                                                   | GT : 78                                                                            | Prediction : 71.42                                                                  |
|  |  |  |
|                                                                                   | GT : 82                                                                            | Prediction : 79.35                                                                  |

Figure 3.7: Qualitative results of the proposed approach on the VSCrowd dataset.

The subsequent chapter elucidates the adaptation methodology employed for the proposed crowd-counting network.

## **4. DOMAIN ADAPTATION WITH GUIDING NETWORK**

### **4.1. Introduction**

Recent studies on crowd counting have achieved promising results using convolutional neural network (CNN) architectures. However, achieving high performance using standard CNN methods continues to be challenging due to large variations in real-world crowd datasets, such as camera angles, resolutions, and crowd densities. The most common issue with these methods is usually performance degradation. This problem is known as domain adaptation, and it is an important and active research area in the field of crowd counting with deep learning. By developing models that are able to adapt to different domains, can be improve the performance of crowd counting algorithms in real-world scenarios. To overcome this difficulty, this chapter proposes a new crowd counting approach that combines three methods: a guide network that guides the crowd counting network and generates normalization parameters in different datasets, a Scale-Adjustable Selection Network that has been successful in crowd counting, and the "Distribution Matching" method as the loss function, which has proven to be more successful than many methods based on Gaussian Correction.

The goal of this approach is to ensure that the trained model also achieves successful results in different datasets. Comprehensive experiments on multiple datasets have been conducted, and a performance comparison of the proposed approach against state-of-the-art methods is presented.

### **4.2. Guiding Network**

At the core of the approach lies the Guiding Network - a Convolutional Neural Network (CNN) trained specifically to estimate the number of people present in an image. When given an image as input, this network generates a density map in which each pixel represents an estimated count of people at that location. Guiding Network architecture comprises several convolutional layers, pooling layers, and fully connected layers that work in concert. Convolutional layers extract features from input

images while pooling layers reduce spatial dimensions while still retaining important information; finally, fully connected layers make final crowd count estimates using extracted features. Guiding Network not only estimates crowd counts but also generates normalization parameters for every dataset, such as scale factors and offsets that normalize input images and output density maps - this ensures the network can accommodate for variance in camera angles, resolutions, and crowd densities between datasets.

### **4.3. Distribution Matching Method**

Distribution Matching serves as a loss function in training the model. It aims to minimize differences between estimated crowd counts and ground truth counts by aligning their distributions as closely as possible.

Adjusting the parameters of both the Guiding Network and Scale-Adaptive Selection Network to minimize Wasserstein distance ensures that estimated crowd counts closely resemble ground truth counts. The Wasserstein distance measures differences in probability distributions; by minimizing it, estimates match ground truth counts more closely. More detailed information on the DM-Count method is provided in Section 3.4.

### **4.4. Scale-Adaptive Selection Network**

Scale-Adaptive Selection Network [43] is a core element of the proposed approach that is specifically designed to deal with variations in crowd scale across images. This separate network is trained to estimate crowd scale in an image, and its output is used to adjust the parameters of the Guiding Network to adapt to its scale in each image.

Scale-Adaptive Selection Network consists of several layers designed to analyze an input image at various scales. It uses multi-scale feature extraction techniques that capture crowd information at various levels of detail; then these extracted features are combined and processed together to estimate crowd scale. More detailed information on the Scale-Adaptive Selection Network is provided in Section 3.3.

## 4.5. Batch Normalization

Batch Normalization is an optimization technique introduced in the seminal paper [19]. The principal aim of this methodology is to accelerate the training process and enhance the performance metrics of deep learning architectures, especially convolutional neural networks (CNNs).

Deep learning algorithms iteratively adjust their internal parameters (e.g., weights) in response to the gradients of a loss function. This iterative learning process can become convoluted and inefficient when the distribution of input activations for internal layers undergoes changes during training, a circumstance known as the internal covariate shift.

Batch Normalization addresses this issue by standardizing the inputs (or activations) of each layer to adhere to a standard Gaussian distribution (zero mean and unit variance). This normalization process is conducted for each mini-batch during each training step, hence the nomenclature "Batch Normalization."

To delve into a more technical understanding, given a mini-batch  $B = \{x_1, x_2, \dots, x_m\}$  of activations, the batch normalization operation unfolds as follows:

- Compute the empirical mean of the batch:

$$\mu_B = \frac{1}{m} \sum_{i=1}^m x_i \quad (4.1)$$

- Compute the empirical variance of the batch:

$$\sigma_B^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2 \quad (4.2)$$

- Normalize each activation:

$$\widehat{x}_1^{(k)} = \frac{x_1^{(k)} - \mu_B^{(k)}}{\sqrt{(\sigma_B^{(k)})^2 + \epsilon}} \quad (4.3)$$

,where  $\epsilon$  is a small constant added to improve numerical stability.

- Apply scale and shift transformations to the normalized activations:

$$y_i^{(k)} = \gamma^{(k)} \widehat{x_i^{(k)}} + \beta^{(k)}, \text{ where } \gamma \text{ and } \beta \text{ are learnable parameters introduced to restore the network's representation power.}$$

Batch Normalization brings numerous advantages to the training process of CNNs and deep learning models:

- **Acceleration of Training:** By mitigating the internal covariate shift, batch normalization enables each layer of the network to learn in an independent manner. This facilitates faster convergence and consequently, a reduction in training time.
- **Regularization Effect:** Given the normalization is performed at each mini-batch, a degree of noise is introduced into the model, which serves as a form of mild regularization, somewhat akin to the dropout technique, and helps prevent overfitting.
- **Improvement of Gradient Flow:** Normalization of activations can help to alleviate the problems associated with vanishing or exploding gradients, which are commonplace in deep networks. This leads to improved learning stability.
- **Facilitation of Higher Learning Rates:** Normalizing inputs to have zero mean and unit variance results in a smoother optimization landscape, which allows for the usage of higher learning rates and hence, accelerates the learning process.

## 4.6. Detailed Analysis of Guiding Network

Figure 4.1 presents the proposed approach, which is a combination of two networks: GuidedBN [4] and SASNet [2]. The network consists of two parts, as shown in Figure 4.1. The crowd counting part uses the SASNet [2], a standard supervised method, to obtain the density map of the input images. During the training of the network, the guiding network learns to estimate the Guided Batch Normalization (GBN) parameters of the relevant scene, and at testing time, the guiding network adjusts the SASNet normalization parameters according to a different testing scene. Drawing inspiration from SASNet, a multi-scale feature vector has been obtained using a multi-level network layer. SASNet consists of three components: First, the

VGG16 base architecture has been used to extract multi-level feature vectors, and these extracted features have been given as input to both the reliability and density header networks, obtaining trust and density maps. Lastly, these two headers have been combined to obtain the final density maps. The same architectural design of SASNet [2] mentioned in Chapter 3 has been used as a crowd counting network.

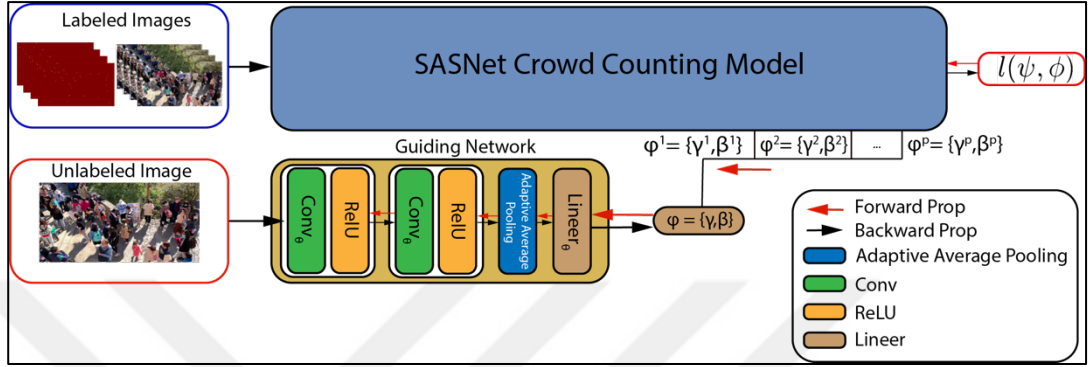


Figure 4.1: General architecture of the proposed crowd counting network.

To elucidate further, one might consider a distinct layer within a Convolutional Neural Network (CNN) architecture and explore the construction of a Guided Batch Normalization (GBN) layer to be integrated after this specific layer. Given a mini-batch comprising  $B$  samples, denoted as  $x_i$ , where  $i$  spans from 1 to  $B$ ,  $x_i$  represents the CNN feature map at the said layer for the  $i$ -th sample. This implies that  $x_i$  belongs to the space  $\mathbb{R}^{H \times W \times D}$ , with  $H \times W$  defining the spatial dimensions and  $D$  representing the channel dimension. Drawing parallels to Batch Normalization (BN) [19], GBN standardizes the activation to ensure zero mean and unit variance along the channel dimension across the mini-batch during the training iteration. Contrasting with BN, which determines the affine transformation parameters  $\gamma$  and  $\beta$  across the samples  $x_i$ :  $i = 1, 2, \dots, B$  in the mini-batch, GBN prognosticates these parameters leveraging the guiding net. The computational process within GBN can be encapsulated as follows:

$$\hat{x}_i[h, w, d] = \text{GBN}(x_i[h, w, d]; \gamma_d, \beta_d), \text{ where}$$

$$\text{GBN} = \gamma_d \cdot ((x_i[h, w, d] - \mu_d)/\sigma_d) + \beta_d$$

$$\forall i \in \{1 \dots B\}, h \in \{1 \dots H\}, w \in \{1 \dots W\}, d \in \{1 \dots D\}$$

Here,  $x_i[h, w, d]$  signifies the  $[h, w, d]$  element of the feature map  $x_i$ , and  $\hat{x}_i[h, w, d]$  represents the normalized output of the activation  $x_i[h, w, d]$ . The mean, denoted as  $\mu_d$ , and the standard deviation, represented as  $\sigma_d$ , are computed from the activation in channel  $d$ , as follows:

$$\mu_d = \frac{\sum_{i=1}^B \sum_{h=1}^H \sum_{w=1}^W x_i[h, w, d]}{B \times H \times W} \quad (4.4)$$

$$\sigma_d = \sqrt{\frac{\sum_{i=1}^B \sum_{h=1}^H \sum_{w=1}^W (x_i[h, w, d] - \mu_d)^2}{B \times H \times W}} + \epsilon \quad (4.5)$$

Within this framework, both  $\mu_d$  and  $\sigma_d$  are scalar values, while  $\epsilon$  is incorporated to maintain numerical robustness. As depicted in Equation 4.4, the affine parameters  $\gamma_d$  and  $\beta_d$ , both scalar entities, dictate the scaling and shifting operations pertinent to the  $d$ -th channel dimension. One might represent the concatenations of  $\{\gamma_d: \forall d\}$  and  $\{\beta_d: \forall d\}$  as  $\gamma$  and  $\beta$  respectively. These variables,  $\gamma$  and  $\beta$ , members of the space  $\mathbb{R}^D$ , serve as the parameters for the GBN layer, with their magnitudes potentially varying across distinct scenes. Broadly, for the  $p$ -th GBN layer, the affine parameters corresponding to the feature channel  $d$  can be denoted as  $\gamma_d^p$  and  $\beta_d^p$ .

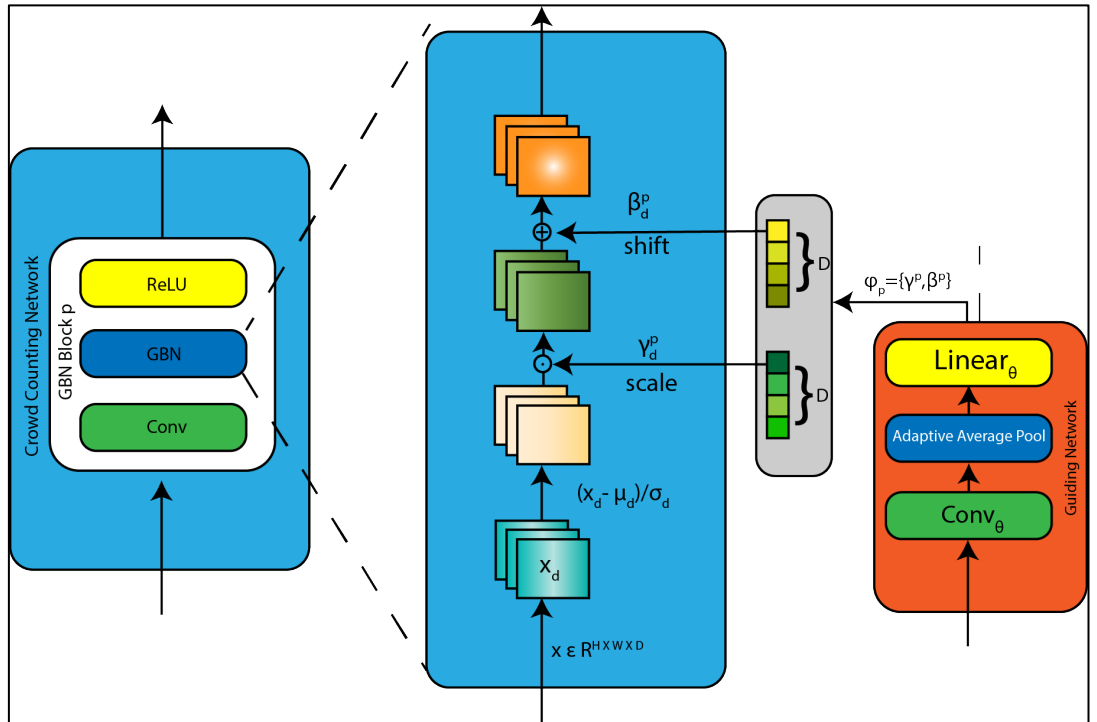


Figure 4.2: Overview of a GBN block in proposed network.

The structure of a GBN block within the defined network architecture is shown in Figure 4.2. Given an input  $x$  to the GBN layer situated in the GBN block  $p$ , the initial step involves normalizing  $x$  across the channel dimensions inherent to the GBN layer. Following this normalization, the affine transformation parameters  $\gamma_p^d$  and  $\beta_p^d$ , corresponding to the  $p$ -th GBN block, are employed to consistently scale and adjust the activation features, culminating in the resultant output  $\hat{x}$ . The generation of these affine parameters is orchestrated by the guiding network, which capitalizes on scene-specific unlabeled data, denoted as  $z$ .

## 4.7. Training Procedure

The proposed model was trained using a combination of Guiding Network, Scale-Adaptive Selection Network and Distribution Matching method training techniques. The Guiding Network guided the crowd counting network and generated normalization parameters across datasets while Scale-Adaptive Selection Network adapted to differing crowd scales across images while Distribution Matching was employed as a loss function to optimize model performance.

Training requires feeding images with their respective ground truth counts into the model while tweaking parameters of networks to minimize loss functions. The validation set serves as an independent, unbiased indicator of its model's progress during training.

The model was trained using a standard backpropagation algorithm with an initial learning rate of 0.0001, conducted over 100 epochs with each update in model parameters using PyTorch [42] on a machine equipped with an NVIDIA RTX 3060 GPU.

## 4.8. Experimental Setup

This research employed an experimental setup designed to rigorously test its proposed crowd counting approach, which uses a Guiding Network, Scale-Adaptive Selection Network and Distribution Matching method as its loss function. Below is more information regarding datasets used, training/testing procedures employed.

## 4.9. Datasets

Experiments were performed using three publicly available crowd counting datasets that provided diverse crowd densities, camera angles and resolutions - such as CityUHK-X [21], FDST [11] and VSCrowd [28] datasets - providing ample testing grounds for the proposed approach. Each of the datasets presented its own distinct challenges when measuring crowd densities or behavioral aspects that allowed for thorough assessment of proposed approach.

The CityUHK-X [21] dataset is composed of 55 camera scenes in total, with 43 scenes forming the training set and the remaining 12 scenes making up the test set. The dimensions of each frame are  $384 \times 512$ .

The VSCrowd [28] Dataset encompasses 62,938 high-resolution surveillance video frames, featuring 2,344,276 annotated pedestrians. It's important to highlight that these video frames were recorded in a variety of environments, such as shopping malls, streets, and scenic areas.

The Fudan-ShanghaiTech [11] dataset is composed of 100 videos captured from 13 different scenes. FDST contains 150,000 frames and 394,081 annotated heads, which makes it larger in terms of the number of frames compared to previous video crowd counting datasets. The training set of the FDST dataset includes 60 videos and 9000 frames, while the test set consists of the remaining 40 videos and 6000 frames. The frames per second (FPS) for FDST is set at 30.

## 4.10. Testing Procedure

The trained model was tested on different datasets. The testing procedure consists of feeding the test images to the trained model and comparing the model's crowd count predictions with ground truth counts provided with the datasets.

The test was performed in a single pass with no parameter updates during the testing phase.

## 4.11. Results

This section outlines the outcomes of experiments carried out to assess the efficacy of the proposed crowd counting method. The performance evaluation is based on two key metrics: Mean Absolute Error (MAE) and Mean Squared Error (MSE).

Table 4.1 shows the test results of the proposed model on the FDST and CityUHK-X datasets respectively after training on the VSCrowd dataset. The results are compared with state-of-the-art methods.

Table 4.1: Domain Adaptation Result over VSCrowd Dataset.

| Model         | VSCrowd $\rightarrow$ FDST |            | VSCrowd $\rightarrow$ CityUHK-X |             |
|---------------|----------------------------|------------|---------------------------------|-------------|
|               | RMSE                       | MAE        | RMSE                            | MAE         |
| CSRNet [29]   | 22.90                      | 19.78      | 29.20                           | 19.26       |
| Adacrowd [45] | 9.7                        | 6.9        | 31.2                            | 21.5        |
| DM-Count [55] | 10.7                       | 7.9        | 30.8                            | 21.5        |
| <b>Our</b>    | <b>4.4</b>                 | <b>3.3</b> | <b>28.3</b>                     | <b>19.2</b> |

According to the results, the proposed method outperformed the other methods both on the FDST dataset and the CityUHK-X dataset.

In the second experiment, the proposed model is trained on the FDST dataset and then tested on the VSCrowd and CityUHK-X datasets respectively. The results are worse than the other methods.

Table 4.2: Domain Adaptation Result over FDST Dataset.

| Model         | FDST $\rightarrow$ VSCrowd |              | FDST $\rightarrow$ CityUHK-X |              |
|---------------|----------------------------|--------------|------------------------------|--------------|
|               | RMSE                       | MAE          | RMSE                         | MAE          |
| CSRNet [29]   | 122.11                     | 84.11        | 34.27                        | 25.78        |
| Adacrowd [45] | 52.16                      | 30.26        | 32.36                        | 21.68        |
| DM-Count [55] | <b>50.35</b>               | <b>22.15</b> | <b>30.17</b>                 | <b>21.72</b> |
| <b>Our</b>    | 61.65                      | 31.73        | 37.03                        | 28.08        |

In the final test, the model was trained on the CityUHK-X dataset and then tested on the VSCrowd and FDST datasets respectively. According to the results, the proposed method outperformed the state-of-the-art methods.

Table 4.3: Domain Adaptation Result over CityUHK-X Dataset.

| Model         | CityUHK-X $\rightarrow$ VSCrowd |              | CityUHK-X $\rightarrow$ FDST |             |
|---------------|---------------------------------|--------------|------------------------------|-------------|
|               | RMSE                            | MAE          | RMSE                         | MAE         |
| CSRNet [29]   | 89.84                           | 44.85        | 7.49                         | 5.78        |
| Adacrowd [45] | 48.79                           | 37.98        | 8.80                         | 6.96        |
| DM-Count [55] | 52.05                           | 35.24        | 41.18                        | 31.80       |
| <b>Our</b>    | <b>48.38</b>                    | <b>23.59</b> | <b>6.31</b>                  | <b>4.91</b> |

According to the results, the models trained with VSCrowd and CityUHK-X gave the best results, while the model trained with the FDST dataset gave worse results.

A key factor behind the successful outcomes of the proposed approach can be attributed to several elements. First, its Guiding Network effectively guides the crowd counting network and generates normalization parameters across different datasets for accurate crowd counting in varied real-world situations. Second, its Scale-Adaptive Selection Network adapts to differing scales of crowds across images for improved estimates of crowd counts estimates. Thirdly, Distribution Matching acts as a loss function by optimizing model performance by reducing differences between estimated and ground truth counts estimates. Although these results are encouraging, there remains room for enhancement. For example, performance could potentially be enhanced by fine-tuning parameters of Guiding Network, Scale-Adaptive Selection Network and Distribution Matching method as well as further optimizing to reduce computation time and resource consumption.

In the following chapter, there is a discussion on experimental studies about image-to-image translation methods as another domain adaptation technique, and the utilization of synthetic datasets.

## 5. DOMAIN ADAPTATION WITH IMAGE-TO-IMAGE TRANSLATION

### 5.1. Introduction

One of the biggest problems in crowd counting is not enough data sets. While a model trained with one data set has high performance, its performance may decrease when tested with another data set. The reasons for this are because mentioned in chapter 4. Different methods are proposed to overcome this problem. One of them is the use of synthetic datasets. However, the desired performance cannot be achieved due to effects such as domain shifting between these datasets and real images. One of the methods that can be used to overcome this problem is the Image to Image(I2I) Translation method. With this method, the style information in the real-world data set is transferred to the synthetic data set, and the domain shifting effect is tried to be eliminated. In this way, aim to contribute to the adaptation of different domains by training in very different environments such as crowd distribution, weather, and light intensity.

The approach introduced in [56] is notable for its application in crowd counting. CCWId [56] developed a large-scale, diverse synthetic dataset, pre-trained a crowd counter on this synthetic data, and fine-tuned it on real data. They proposed a counting method that utilizes a domain adaptation-based cycle GAN, thereby relieving humans from the burden of extensive data annotations.

In this section, the analyzed the adaptation process from a source domain to a target domain using the network model proposed in Section 3 and state-of-the-art image-to-image translation methods. GCC synthetic dataset is used as the source domain. The unlabeled CityUHK-X dataset is selected as the target domain. The aim is to make the labeled synthetic dataset like the unlabeled CityUHK-X dataset using image-to-image translation methods and to train the proposed network with the resulting labeled dataset to improve the performance in the target domain. Figure 5.1 represents the workflow mentioned above. In this section, the analyzed the adaptation process from a source domain to a target domain using the network model proposed in Section 3 and state-of-the-art image-to-image translation methods.

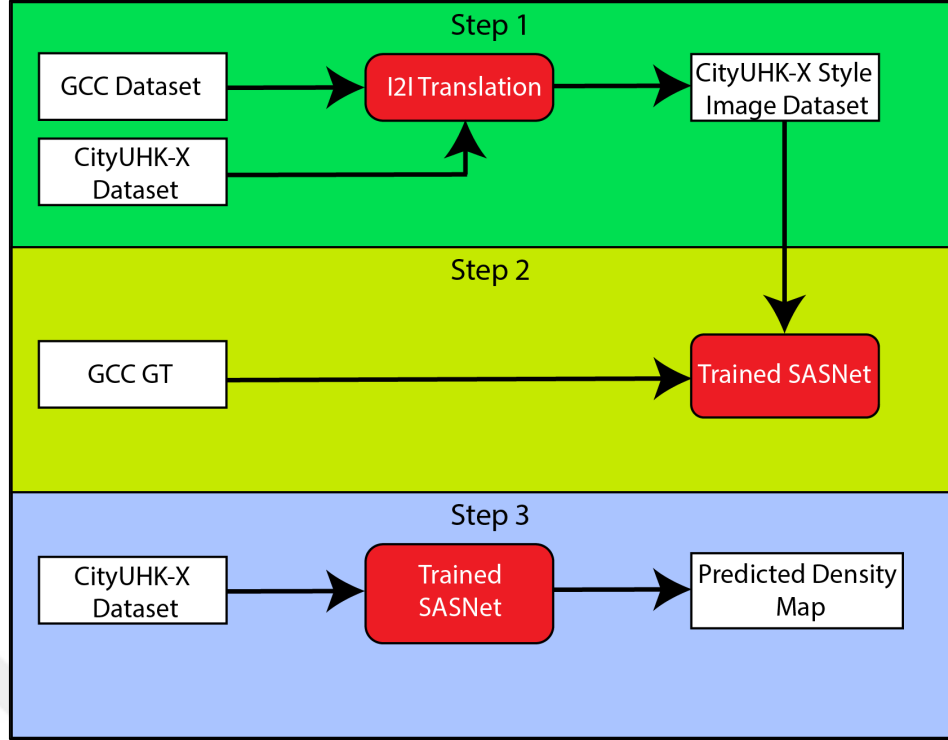


Figure 5.1: Image-To-Image Translation Workflow.

According to the workflow presented, there is first the GCC dataset, which is labeled, and the CityUHK-X datasets, which are unlabeled. The objective is to make the labeled GCC dataset resemble the unlabeled CityUHK-X dataset. This is done with the intention of ensuring that the model trained with the GCC dataset performs successfully on the CityUHK-X dataset. Initially, images from both the GCC and CityUHK-X datasets are input into the I2I Translation GAN model, producing a new dataset that retains the labels of the GCC dataset but adopts the style of the CityUHK-X dataset. In the subsequent step, this new dataset, along with the GTs from the GCC dataset, are input into the model introduced in Section 3 for training. Finally, in Section 3, the newly trained model undergoes testing using the CityUHK-X test dataset.

The subsequent section provides a brief overview of CycleGAN [72], which serves as an example of an I2I Translation method.

## 5.2. CycleGAN

Zhu and colleagues [72], proposed a technique that leverages the concept of Generative Adversarial Networks (GANs) to learn how to transform an image from a source domain to a target domain, eliminating the need for paired examples. To further

regulate the mappings, they introduced two-cycle consistency losses. This concept is based on the idea that if translated from one domain to another and then back again, it should return to the original starting point. The CycleGAN architecture comprises two GANs - a generator and a discriminator, as depicted in the figure below. These two GANs operate concurrently to generate two types of images: those in the style of the source domain and those in the style of the target domain. Figure 5.2 shows the general structure of the CycleGAN approach.

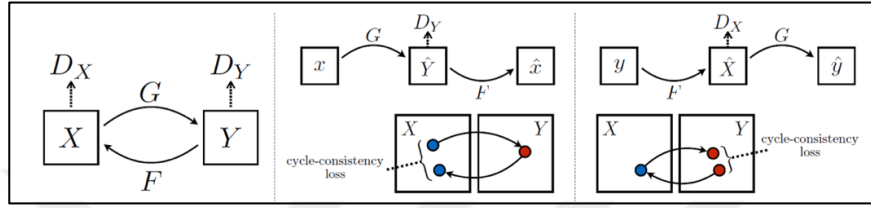


Figure 5.2: General structure of CycleGAN approach.

### 5.3. Training Procedure

In the formulation of the novel dataset, images originating from both the source and target datasets were subjected to a standardization process to achieve uniform dimensions. The CityUHK-X dataset's dimensions were employed as the benchmark, necessitating the adaptation of the GCC synthetic dataset to conform to this established standard. To elucidate, images possessing dimensions of  $1920 \times 1080$  were partitioned into sub-images with dimensions of  $512 \times 384$ . Comprehensive details pertaining to the resultant datasets are delineated in Table 5.1.

Table 5.1: Image-to-Image Translation Datasets.

| Dataset       | Image Size       | Image Count |      |
|---------------|------------------|-------------|------|
|               |                  | Train       | Test |
| CityUHK-X[21] | $512 \times 384$ | 2503        | 687  |
| GCC[56]       | $512 \times 384$ | 7920        | 2838 |

A representative illustration of the segmentation process applied to the images within the GCC dataset is presented in Figure 5.4. In this context, the image of dimensions  $1920 \times 1080$ , as depicted in Figure 5.3, undergoes division into sub-images of  $512 \times 384$  dimensions utilizing the image partitioning algorithm. Owing to the

incongruence between the original image dimensions and the desired 512x384 segments, each resultant sub-image exhibits overlapping regions with its adjacent counterparts.



Figure 5.3 Sample GCC Image.



Figure 5.4 Sample GCC Image After Splitting.

Figure 5.5 shows the visual results of the transformation from the GCC dataset to the CityUHK-X dataset using the CyleGAN [72] method.



Figure 5.5: GCC to CityUHK-X Translation Results(CycleGAN).

Figure 5.6 shows the visual results of the transformation from the GCC dataset to the CityUHK-X dataset using the DiscoGAN [22] method.



Figure 5.6: GCC to CityUHK-X Translation Results(DiscoGAN).

Figure 5.7 shows the visual results of the transformation from the GCC dataset to the CityUHK-X dataset using the S-CycleGAN [56] method.



Figure 5.7: GCC to CityUHK-X Translation Results(S-CycleGAN).

Figure 5.8 shows the visual results of the transformation from the GCC dataset to the CityUHK-X dataset using the DualGAN [63] method.



Figure 5.8: GCC to CityUHK-X Translation Results(DualGAN).

## 5.4. Results

In the subsequent subsection, results are elucidated pertaining to a variety of GAN-centric methodologies that facilitate image-to-image translation. Table 5.2 provides a comprehensive delineation of the test outcomes, which were derived from the training of the crowd counting network using datasets generated by distinct GAN paradigms. Within this framework, the term 'lower baseline' is utilized to denote outcomes that were trained on the GCC dataset but subsequently tested on the CityUHK-X dataset. In contrast, the 'upper baseline' is indicative of results where both the training and testing phases were exclusively conducted on the CityUHK-X dataset.

Table 5.2: Domain Adaptation Results for GCC to CityUHK-X Scenario.

| Methods        | RMSE        | MAE         |
|----------------|-------------|-------------|
| Lower Baseline | 103.3       | 28.8        |
| CycleGAN[72]   | <b>38.1</b> | <b>25.8</b> |
| DualGan[63]    | 52.75       | 41.36       |
| DiscoGAN[22]   | 146.98      | 111.67      |
| S-CycleGAN[56] | 120.13      | 106.8       |
| Upper Baseline | 12.55       | 8.24        |

According to the results, only CyleGAN showed improvement compared to Lower Baseline. Since image-to-image translation is not the main focus of this thesis, no new architecture is proposed. Instead, possible improvements of existing methods

on the problem of space adaptation in crowd counting are presented. While the preliminary findings are encouraging, there's potential for superior outcomes in this domain by re-evaluating Image-to-Image translation techniques tailored for crowd counting. As a future work in this area, new specialized loss methods for crowd counting can be proposed for image-to-image GAN methods. For example, the image produced by GAN can be used as input by a crowd counting network and the loss between the estimated density map produced as output and the ground truth can be calculated and used in the GAN fusion function. Thus, better performance can be achieved by using the loss between the output density maps instead of the directly generated image.

In the subsequent chapter, there is a discussion regarding the potential implications of incorporating an adaptive kernel feature to enhance the efficacy of the Crowded Scale-Adaptive Selection Network.

## 6. ADAPTIVE KERNEL

### 6.1. Introduction

Convolutional Neural Networks (CNN) are widely utilized for crowd counting. CNNs use kernels (or filters) of various sizes to learn the local features of an image; however, often these sizes remain static, which limits their capacity to recognize features at specific scales. An adaptive CNN model may therefore enable learning features at multiple scales simultaneously. Adaptive kernel sizes enable models to dynamically change their kernel sizes during the learning process, helping it more efficiently recognize features at various scales; for instance, smaller kernel sizes could help identify small objects while larger kernel sizes would detect larger objects in images. To create a CNN model with adaptive kernel sizes, a mechanism must exist that allows the model to adapt its kernel sizes during learning; typically, this is achieved using some form of learning algorithm or optimization technique.

A Study [53] have demonstrated the advantages of employing CNN models with adjustable kernel sizes on tasks like image classification and object detection, due to their ability to learn features at varying scales. In this paper introduces a differentiable, backpropagation-trainable Gaussian envelope for kernel size that can grow or shrink on a base grid. Its coefficients are calculated using normalized grid coordinates with current aperture to produce envelope coefficients; then multiplied element-wise with weights prior to convolution.

The paper also discusses the significance of initializing weights and envelope parameters in a neural network, suggesting that weight variance should be adjusted so that layer inputs and outputs have equal variance. Unfortunately, adaptive kernels scale weights with envelope coefficients that alter propagated signals; to mitigate this problem this paper proposes using element-wise multiplication to normalize and scale envelope coefficients so as to maintain unchanged variance weights via element multiplication operation. Figure 5.1 shows the working diagram of the proposed method.

In this section, the aim is to improve performance by modifying some convolution layers used in the crowd-counting network on the crowd-counting

network architecture proposed in chapters 3 and 4 with the adaptive kernel method described in [53]

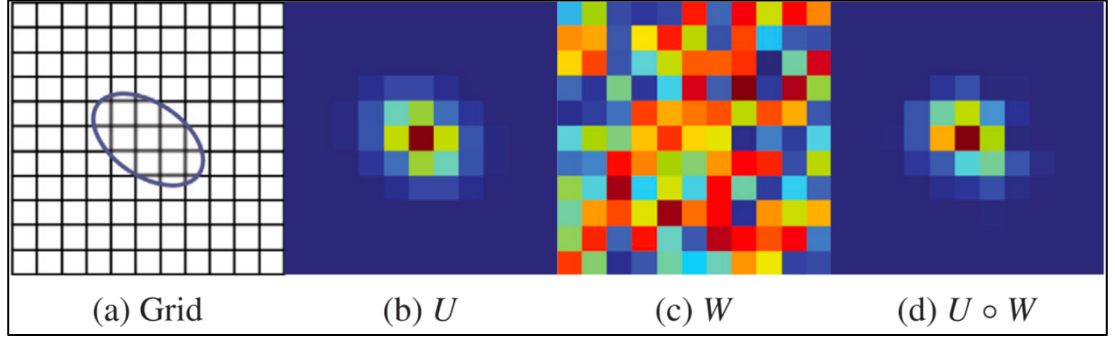


Figure 6.1: Illustration of the weight envelope.

Figure 6.1 shows the use of the weight envelope mentioned in [53]. First, (a) shows an arbitrary differentiable envelope function on a base grid. In (b) shown an example Gaussian envelope ( $U$ ), in (c) shown the randomly generated weights ( $W$ ), and finally in (d) the weights masked by the Gaussian envelope shown in (b) are shown.

## 6.2. Detailed Analysis of Adaptive Kernel Architecture

This method for training the size of a convolutional kernel using a Gaussian envelope that controls its size; backpropagation allows dynamic adjustment of this envelope so that its control over kernel size can be managed dynamically. This method proposed by [53] is described in more detail below.

In the proposed adaptive model, the kernel size is controlled by a Gaussian envelope function, which is a function that gives the weights of the kernel. This envelope function is differentiable and trainable by backpropagation, which allows the kernel size to be dynamically adjusted during training. The parameter  $\Sigma_u$  determines the width of the Gaussian envelope that controls the size of the kernel.

During the feed-forward phase, the envelope function, denoted as  $U_f$ , is calculated based on the normalized grid coordinates, leading to the generation of envelope coefficients, represented as  $U_k$  and  $l$ . These coefficients are then subjected to element-wise multiplication with the weights prior to the convolution process. During the training phase, it's crucial to monitor and limit the  $\Sigma_u$  value to ensure it remains

above a certain threshold (for instance,  $1/n$ ) to avoid excessive shrinking caused by high gradient values and potential fluctuations. This is of significance because if  $\Sigma_u$  becomes excessively small, the kernel may become overly narrow, thereby failing to capture the necessary features in the input data. In summary, the training of the kernel size in this method involves training the  $\Sigma_u$  parameter that regulates the size of the kernel, computing the envelope function during the feed-forward phase, and monitoring and limiting the  $\Sigma_u$  value during the training phase to prevent excessive shrinking. [53]

During the feed-forward operation, the envelope function, denoted as  $U_f$ , is calculated based on normalized grid coordinates, resulting in the generation of envelope coefficients, represented as  $U_k$  and  $l$ . Subsequently, these coefficients are element-wise multiplied with the weights before the convolution process is carried out.

Initializing weights is crucial to increasing training and generalization capacity of neural networks, yet initializing weights with adaptive kernels complicates matters as their envelope coefficients alter weight scaling while changing propagated signals' variance, making it uncertain what value to use when calculating variance of the weights; nevertheless, an appropriate initialization variance for weights in presence of envelope can still be determined.

There are no official guidelines regarding initializing envelope parameters; other than specifying they must be positive and non-zero, initializing  $S_u$  should fall between  $[1/n, n]$  for initialization to work optimally during training (kernel size =  $n$ ). Nonetheless, during this process monitoring and clipping must take place so as to keep its value above certain values (such as  $1/n$ ) so as to avoid over shrinkage due to high gradient values or fluctuations that might occur.

Figure 6.2 provides a pseudo-code description of the details described above.

```

1: procedure INITIALIZE(X, in_channels, out_channels, kernel_size, stride,
padding, dilation, groups, use_bias)
2:    $W \leftarrow$  initialize with size (out_channels, in_channels, kernel_size, kernel_size)
3:    $\sigma \leftarrow$  initialize with size (out_channels)
4:    $\mu \leftarrow [0.5, 0.5]$ 
5: end procedure
6: procedure FORWARD(X)
7:    $U \leftarrow$  _CALC_U
8:    $adaptive\_weights \leftarrow W \times U$ 
9:   Convolve X using  $adaptive\_weights$ 
10:  if use_bias is True then
11:    Add bias to the output
12:  end if
13:  Apply batch normalization
14:  Apply ReLU activation
15:  return output
16: end procedure
17: procedure _CALC_U
18:   $up \leftarrow \sum_i (idxs - \mu)^2$ 
19:   $dwn \leftarrow 2(\sigma^2) + 1e - 8$ 
20:   $result \leftarrow \exp\left(-\frac{up}{dwn}\right)$ 
21:  Reshape result to kernel size
22:  if normalization is applied then
23:    Normalize masks
24:  end if
25:  return masks
26: end procedure

```

Figure 6.2: Adaptive Kernel Workflow.

In the study, the proposed adaptive kernel approach is added to the first layers in the decoding block of the SASNet architecture presented in Chapter 3. The reason for adding the Adaptive kernel layer to the first layer of the decoder block is that it is the first layer that meets the feature maps and it is thought that an improvement that can be made here will have a greater effect on the output of the network. Figure 6.3 shows the location of the adaptive CNN layers in the decoding block.

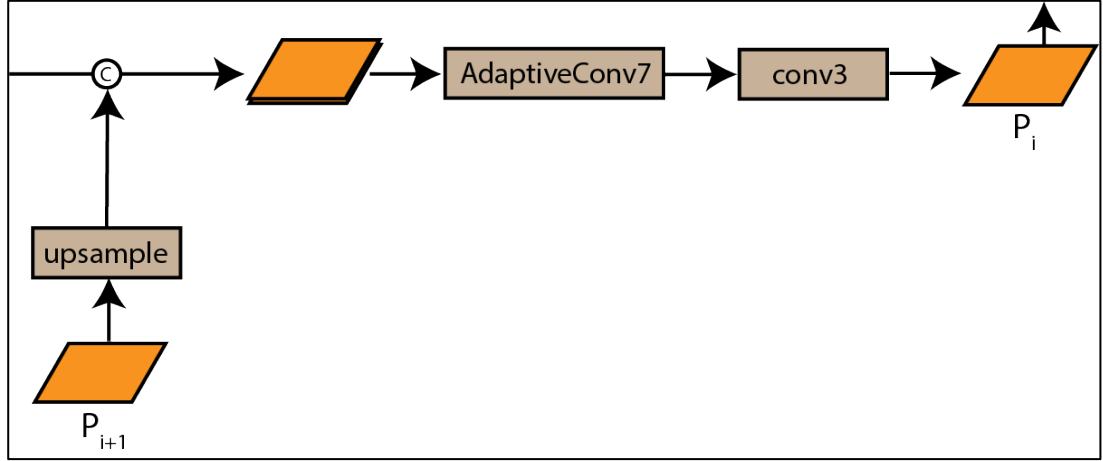


Figure 6.3: Use of Adaptive Convolution in the decoding block on the proposed architecture.

### 6.3. Testing Procedure

The trained model was tested on different datasets. The testing procedure consists of feeding the test images to the trained model and comparing the model's crowd count predictions with ground truth counts provided with the datasets.

The test was performed in a single pass with no parameter updates during the testing phase.

### 6.4. Results

This chapter delineates the outcomes of experiments undertaken to assess the efficacy of the proposed crowd counting methodology. The evaluation metrics employed are Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE).

Table 6.1 provides a comprehensive view of the test outcomes for the proposed model when applied to the FDST and CityUHK-X datasets. These results are juxtaposed with those of contemporary state-of-the-art techniques.

Table 6.1: Adaptive Kernel Results.

| Model         | CityUHK-X[21] |            | FDST[11]    |             |
|---------------|---------------|------------|-------------|-------------|
|               | RMSE          | MAE        | RMSE        | MAE         |
| CSRNet [29]   | 16.4          | 10.9       | 5.27        | 4.04        |
| Adacrowd [45] | 16.1          | 10.8       | 3.1         | 2.4         |
| RAZNet [31]   | -             | -          | 5.5         | 4.7         |
| STANet [58]   | -             | -          | 5.1         | 3.6         |
| EPF [33]      | 15.38         | 11.07      | 2.6         | 2.2         |
| LST [11]      | -             | -          | 4.5         | 3.4         |
| DM-Count [55] | 27.39         | 19.44      | <b>2.45</b> | <b>1.81</b> |
| SFCN [56]     | 13.8          | 9.7        | 2.5         | 1.9         |
| <b>Our</b>    | <b>12.9</b>   | <b>8.4</b> | 8           | 5.8         |

According to the results, the model trained with CityUHK-X gave the best results, while the model trained with the FDST dataset gave worse results. However, the results obtained were no better than the results obtained in Chapter 3. One of the reasons for this may be that the algorithm does not work well enough or a bug in the Pytorch implementation of the method. Or you can try different sections on the adaptive layer decoder blog. As a result, this area is open to improvement and better results can be obtained with future work.

## 7. CONCLUSIONS AND FUTURE WORKS

This thesis presents a novel approach to crowd counting that leverages the strengths of three methods - Guiding Network [45], Scale-Adaptive Selection Network [43] and Distribution Matching [55] method as loss function - simultaneously. This multi-method approach addresses some of the limitations inherent in traditional crowd counting methods while offering more efficient solutions for estimating people present.

The proposed approach has shown remarkable success in experiments, outshone standard Convolutional Neural Network (CNN) methods in terms of accuracy and robustness. The Guiding Network successfully guides the crowd counting network while creating normalization parameters across datasets for more accurate counting in diverse real-world situations. Furthermore, Scale-Adaptive Selection Network has also been utilized successfully while the Distribution Matching method proved superior than many Gaussian-based methods as a loss function.

In addition, studies were carried out on the use of image-to-image translation methods in crowd counting to solve the domain adaptation problem in the study, and a simulation study was made with real world datasets by using the GAN-based state-of-the-art methods and synthetic dataset in the literature. The aim here is to use the advantages of synthetically overlaid datasets and to find a solution to the problem of domain adaptation by comparing these images to real-world datasets. The studies have been promising and open the door to future studies.

Finally, aimed to achieve more successful results by using the adaptive kernel method to make the crowd counting networks more dynamic and adaptive, by enabling the network to learn the most appropriate kernel size in the relevant CNN layer during the training phase.

Despite the encouraging results, there remain several avenues of research for further development. One potential route could be exploring other networks or methods that could enhance the proposed approach further by being combined with it; such as other types of neural networks or machine learning algorithms being employed instead or in addition to CNNs in the proposed approach.

As a future work about the image-to-image translation for crowd counting area, new specialized loss methods for crowd counting can be proposed for the image-to-

image translation GAN methods. For example, the image produced by GAN can be used as input by a crowd-counting network and the loss between the estimated density map produced as output and the ground truth can be calculated and used in the GAN fusion function. Thus, better performance can be achieved by using the loss between the output density maps instead of the directly generated image.

Future research could investigate adapting the proposed method for other related tasks, like crowd counting. Although the proposed method has proven itself successful with crowd counts, it could also prove effective for other purposes that require counting objects such as object detection or tracking. Future research could also focus on increasing the efficiency of the proposed approach. Though it has already proven more accurate than traditional methods, further optimizations may make it even more efficient with regards to computation time or resource consumption.

This thesis presents a new approach to crowd counting that addresses the limitations of traditional methods, providing a more effective solution. While there is still much work to be done on crowd counting technology, the proposed approach is believed to represent a significant advance.

## REFERENCES

- [1] Ali S., Shah M., (2008), “Floor Fields for Tracking in High Density Crowd Scenes” Computer Vision – ECCV 2008, 1–14, Berlin, Heidelberg, Germany, October.
- [2] Bai H., Mao J., Gary Chan S.-H., (2022), “A survey on deep learning-based single image crowd counting: Network design, loss function and supervisory signal” *Neurocomputing*, 508, 1–18.
- [3] Cai Y., Chen L., Ma Z., Lu C., Wang C., He G., (2021), “Leveraging Intra-Domain Knowledge to Strengthen Cross-Domain Crowd Counting” 2021 IEEE International Conference on Multimedia and Expo (ICME), 1–6, Shenzhen, China, July.
- [4] Chan A.B., Zhang-Sheng John Liang, Vasconcelos N., (2008), “Privacy preserving crowd monitoring: Counting people without people models or tracking” 2008 IEEE Conference on Computer Vision and Pattern Recognition, 1–7, Anchorage, AK, USA, June.
- [5] Chen B., Yan Z., Li K., Li P., Wang B., Zuo W., Zhang L., (2021), “Variational Attention: Propagating Domain-Specific Knowledge for Multi-Domain Learning in Crowd Counting” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 16045–16055, Montreal, QC, Canada, October.
- [6] Chen G.J., (2012), “A simple way to deal with multicollinearity” *J. Appl. Stat.*, 39(9), 1893–1909.
- [7] Cong Zhang, Hongsheng Li, Wang X., Xiaokang Yang, (2015), “Cross-scene crowd counting via deep convolutional neural networks” 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 833–841, Boston, MA, USA, June.
- [8] Cuturi M., (2013), “Sinkhorn Distances: Lightspeed Computation of Optimal Transport” *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, 2292–2300, Red Hook, NY, USA, December.
- [9] Dalal N., Triggs B., (2005), “Histograms of Oriented Gradients for Human Detection” 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), 886–893, San Diego, CA, USA, June.
- [10] Dumoulin V., Visin F., (2016), “A guide to convolution arithmetic for deep learning” *ArXiv E-Prints*, arXiv:1603.07285.
- [11] Fang Y., Zhan B., Cai W., Gao S., Hu B., (2019), “Locality-Constrained Spatial Transformer Network for Video Crowd Counting” 2019 IEEE International Conference on Multimedia and Expo (ICME), 814–819, Shanghai, China, July.

- [12] Felzenszwalb P.F., Girshick R.B., McAllester D., Ramanan D., (2010), “Object Detection with Discriminatively Trained Part-Based Models” IEEE Trans. Pattern Anal. Mach. Intell., 32(9), 1627–1645.
- [13] Gao J., Han T., Yuan Y., Wang Q., (2023), “Domain-Adaptive Crowd Counting via High-Quality Image Translation and Density Reconstruction” IEEE Trans. Neural Netw. Learn. Syst., 34(8), 4803–4815.
- [14] Gao J., Yuan Y., Wang Q., (2021), “Feature-Aware Adaptation and Density Alignment for Crowd Counting in Video Surveillance” IEEE Trans. Cybern., 51(10), 4822–4833.
- [15] Gong S., Zhang S., Yang J., Dai D., Schiele B., (2022), “Bi-level Alignment for Cross-Domain Crowd Counting” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7532–7540, New Orleans, LA, USA, June.
- [16] He K., Zhang X., Ren S., Sun J., (2016), “Deep Residual Learning for Image Recognition” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770–778, Las Vegas, NV, USA, June.
- [17] He Y., Ma Z., Wei X., Hong X., Ke W., Gong Y., (2021), “Error-Aware Density Isomorphism Reconstruction for Unsupervised Cross-Domain Crowd Counting” Proc. AAAI Conf. Artif. Intell., 35(2), 1540–1548.
- [18] Idrees H., Saleemi I., Seibert C., Shah M., (2013), “Multi-source Multi-scale Counting in Extremely Dense Crowd Images” 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2547–2554, Portland, OR, USA, June.
- [19] Ioffe S., Szegedy C., (2015), “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift” Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, 448–456, Lille, France, June.
- [20] Jaderberg M., Simonyan K., Zisserman A., Kavukcuoglu K., (2015), “Spatial Transformer Networks” Adv. Neural Inf. Process. Syst., 28, 2017–2025.
- [21] Kang D., Dhar D., Chan A., (2017), “Incorporating Side Information by Adaptive Convolution” Advances in Neural Information Processing Systems, 3870–3880, Red Hook, NY, USA, December.
- [22] Kim T., Cha M., Kim H., Lee J.K., Kim J., (2017), “Learning to Discover Cross-Domain Relations with Generative Adversarial Networks” Proceedings of the 34th International Conference on Machine Learning, 1857–1865, Sydney, NSW, Australia, August.
- [23] Kratz L., Nishino K., (2009), “Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models” 2009 IEEE Conference on Computer Vision and Pattern Recognition, 1446–1453, Miami, FL. USA, June.

- [24] Krishna Reddy M.K., Asiful Hossain M., Rochan M., Wang Y., (2020), “Few-Shot Scene Adaptive Crowd Counting Using Meta-Learning” 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), 2803–2812, Snowmass Village, CO, USA, March.
- [25] Krizhevsky A., Sutskever I., Hinton G.E., (2017), “ImageNet classification with deep convolutional neural networks” *Commun. ACM*, 60(6), 84–90.
- [26] LeCun Y., Bengio Y., Hinton G., (2015), “Deep learning” *Nature*, 521(7553), 436–444.
- [27] Lecun Y., Bottou L., Bengio Y., Haffner P., (1998), “Gradient-based learning applied to document recognition” *Proc. IEEE*, 86(11), 2278–2324.
- [28] Li H., Liu L., Yang K., Liu S., Gao J., Zhao B., Zhang R., Hou J., (2022), “Video Crowd Localization With Multifocus Gaussian Neighborhood Attention and a Large-Scale Benchmark” *IEEE Trans. Image Process.*, 31, 6032–6047.
- [29] Li Y., Zhang X., Chen D., (2018), “CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1091–1100, Salt Lake City, UT, June.
- [30] Litjens G., Kooi T., Bejnordi B.E., Setio A.A.A., Ciompi F., Ghafoorian M., Van Der Laak J.A.W.M., Van Ginneken B., Sánchez C.I., (2017), “A survey on deep learning in medical image analysis” *Med. Image Anal.*, 42, 60–88.
- [31] Liu C., Weng X., Mu Y., (2019), “Recurrent Attentive Zooming for Joint Crowd Counting and Precise Localization” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 1217–1226, Long Beach, CA, USA, June.
- [32] Liu W., Durasov N., Fua P., (2022), “Leveraging Self-Supervision for Cross-Domain Crowd Counting” 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5331–5342, New Orleans, LA, USA, June.
- [33] Liu W., Salzmann M., Fua P., (2020), “Estimating People Flows to Better Count Them in Crowded Scenes” *Computer Vision – ECCV 2020*, 723–740, Cham, August.
- [34] Liu W., Salzmann M., Fua P., (2019), “Context-Aware Crowd Counting” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 5094–5103, Long Beach, CA, USA, June.
- [35] Loy C.C., Chen K., Gong S., Xiang T., (2013), “Crowd Counting and Profiling: Methodology and Evaluation” *Modeling, Simulation and Visual Analysis of Crowds*, 347–382, New York, NY, October.
- [36] Loy C.C., Xiang T., Gong S., (2012), “Salient motion detection in crowded scenes” 2012 5th International Symposium on Communications, Control and Signal Processing, 1–4, Roma, Italy, May.

- [37] Loy C.C., Xiang T., Gong S., (2010), “Time-Delayed Correlation Analysis for Multi-Camera Activity Understanding” *Int. J. Comput. Vis.*, 90(1), 106–129.
- [38] Ma Z., Hong X., Wei X., Qiu Y., Gong Y., (2021), “Towards A Universal Model for Cross-Dataset Crowd Counting” 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 3185–3194, Montreal, QC, Canada, October.
- [39] Mehran R., Oyama A., Shah M., (2009), “Abnormal crowd behavior detection using social force model” 2009 IEEE Conference on Computer Vision and Pattern Recognition, 935–942, Miami, FL, June.
- [40] Nair V., Hinton G.E., (2010), “Rectified Linear Units Improve Restricted Boltzmann Machines” *Proceedings of the 27th International Conference on International Conference on Machine Learning*, 807–814, Madison, WI, USA, June.
- [41] Paragios N., Ramesh V., (2001), “A MRF-based approach for real-time subway monitoring” *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, I-1034–I-1040, Kauai, HI, USA, December.
- [42] Paszke A., Gross S., Chintala S., Chanan G., Yang E., DeVito Z., Lin Z., Desmaison A., Antiga L., Lerer A., (2017), “Automatic differentiation in PyTorch” 31st Conference on Neural Information Processing Systems (NIPS 2017), 1–4, Long Beach, California, USA, December.
- [43] Qingyu Song, Chang-An Wang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Jian Wu, Jiayi Ma, (2021), “To Choose or to Fuse? Scale Selection for Crowd Counting” *Proc. AAAI Conf. Artif. Intell.*, 35(3), 2576–2583.
- [44] Ranjan V., Le H., Hoai M., (2018), “Iterative crowd counting” *Proceedings of the European Conference on Computer Vision (ECCV)*, 270–285, Munich, Germany, September.
- [45] Reddy M.K.K., Rochan M., Lu Y., Wang Y., (2022), “AdaCrowd: Unlabeled Scene Adaptation for Crowd Counting” *IEEE Trans. Multimed.*, 24, 1008–1019.
- [46] Rodriguez M., Laptev I., Sivic J., Audibert J.-Y., (2011), “Density-aware person detection and tracking in crowds” 2011 International Conference on Computer Vision, 2423–2430, Barcelona, Spain, November.
- [47] Sam D.B., Peri S.V., Sundararaman M.N., Kamath A., Babu R.V., (2021), “Locate, Size, and Count: Accurately Resolving People in Dense Crowds via Detection” *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(8), 2739–2751.
- [48] Sam D.B., Surya S., Babu R.V., (2017), “Switching Convolutional Neural Network for Crowd Counting” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 4031–4039, Honolulu, HI, July.
- [49] Schmidhuber J., (2015), “Deep learning in neural networks: An overview” *Neural Netw.*, 61(1), 85–117.

- [50] Sermanet P., Eigen D., Zhang X., Mathieu M., Fergus R., LeCun Y., (2014), “OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks” 2nd International Conference on Learning Representations (ICLR 2014), 1–16, Banff, AB, Canada, April.
- [51] Sreenu G., Saleem Durai M.A., (2019), “Intelligent video surveillance: a review through deep learning techniques for crowd analysis” *J. Big Data*, 6(1), 48.
- [52] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R., (2014), “Dropout: A Simple Way to Prevent Neural Networks from Overfitting” *J Mach Learn Res*, 15(1), 1929–1958.
- [53] Tek F.B., Çam İ., Karlı D., (2021), “Adaptive convolution kernel for artificial neural networks” *J. Vis. Commun. Image Represent.*, 75(1), 103015.
- [54] Tian Y., Brown L., Hampapur A., Lu M., Senior A., Shu C., (2008), “IBM smart surveillance system (S3): event based video surveillance system with an open and extensible framework” *Mach. Vis. Appl.*, 19(5–6), 315–327.
- [55] Wang B., Liu H., Samaras D., Nguyen M.H., (2020), “Distribution Matching for Crowd Counting” *Adv. Neural Inf. Process. Syst.*, 33(1), 1595–1607.
- [56] Wang Q., Gao J., Lin W., Yuan Y., (2019), “Learning From Synthetic Data for Crowd Counting in the Wild” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8190–8199, Long Beach, CA, USA, June.
- [57] Wen L., Du D., Zhu P., Hu Q., Wang Q., Bo L., Lyu S., (2021), “Detection, Tracking, and Counting Meets Drones in Crowds: A Benchmark” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 7808–7817, Nashville, TN, USA, June.
- [58] Wen L., Du D., Zhu P., Hu Q., Wang Q., Bo L., Lyu S., (2019), “Drone-based Joint Density Map Estimation, Localization and Tracking with Space-Time Multi-Scale Attention Network” *CoRR*, abs/1912.01811.
- [59] Wu Q., Wan J., Chan A.B., (2021), “Dynamic Momentum Adaptation for Zero-Shot Cross-Domain Crowd Counting” *Proceedings of the 29th ACM International Conference on Multimedia*, 658–666, Virtual Event China, October.
- [60] Xiong H., Lu H., Liu C., Liu L., Cao Z., Shen C., (2019), “From Open Set to Closed Set: Counting Objects by Spatial Divide-and-Conquer” 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 8361–8370, Seoul, Korea (South), October.
- [61] Yan Z., Li P., Wang B., Ren D., Zuo W., (2021), “Towards Learning Multi-domain Crowd Counting” *IEEE Trans. Circuits Syst. Video Technol.*, 2481–2495.
- [62] Yang S.-D., Su H.-T., Hsu W.H., Chen W.-C., (2021), “Class-agnostic Few-shot Object Counting” 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 869–877, Waikoloa, HI, USA, January.

- [63] Yi Z., Zhang H., Tan P., Gong M., (2017), “DualGAN: Unsupervised Dual Learning for Image-to-Image Translation” 2017 IEEE International Conference on Computer Vision (ICCV), 2868–2876, Venice, October.
- [64] Yukun Tian, Yiming Lei, Junping Zhang, James Z. Wang, (2020), “PaDNet: Pan-Density Crowd Counting” IEEE Trans. Image Process., 29(3), 2714–2727.
- [65] Zeiler M.D., Fergus R., (2014), “Visualizing and Understanding Convolutional Networks” Computer Vision – ECCV 2014, 818–833, Cham, September.
- [66] Zhan B., Monekosso D.N., Remagnino P., Velastin S.A., Xu L.-Q., (2008), “Crowd analysis: a survey” Mach. Vis. Appl., 19(5–6), 345–357.
- [67] Zhang Q., Chan A.B., (2019), “Wide-Area Crowd Counting via Ground-Plane Density Maps and Multi-View Fusion CNNs” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 8289–8298, Long Beach, CA, USA, June.
- [69] Zhang Y., Zhou D., Chen S., Gao S., Ma Y., (2016), “Single-Image Crowd Counting via Multi-Column Convolutional Neural Network” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 589–597, Las Vegas, NV, USA, June.
- [70] Zheng L., Li Y., Mu Y., (2021), “Learning Factorized Cross-View Fusion for Multi-View Crowd Counting” 2021 IEEE International Conference on Multimedia and Expo (ICME), 1–6, Shenzhen, China, July.
- [71] Zhou B., Wang X., Tang X., (2011), “Random field topic model for semantic region analysis in crowded scenes from tracklets” CVPR 2011, 3441–3448, Colorado Springs, CO, USA, June.
- [72] Zhu J.-Y., Park T., Isola P., Efros A.A., (2017), “Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks” 2017 IEEE International Conference on Computer Vision (ICCV), 2242–2251, Venice, October.

## **BIOGRAPHY**

Cengizhan Haldız received his bachelor's degree from the Faculty of Computer and Information Sciences Department of Computer Engineering at Sakarya University in 2017. He is currently a master's student in the Department of Computer Engineering at Gebze Technical University, Turkey. He has been working as a Full Stack Developer in the industry since 2018.



## APPENDICES

### Appendix A: The Publications about the Thesis

Haldız C., Ismael.S.F., Çelebi H., Aptoula E. (2023), “Crowd Counting via Joint SASNet and a Guided Batch Normalization Network”, 31st IEEE Signal Processing and Communications Applications Conference (SIU 2023) İstanbul, Turkey, 5-8 July.



## Appendix B: VGG-16 Architecture

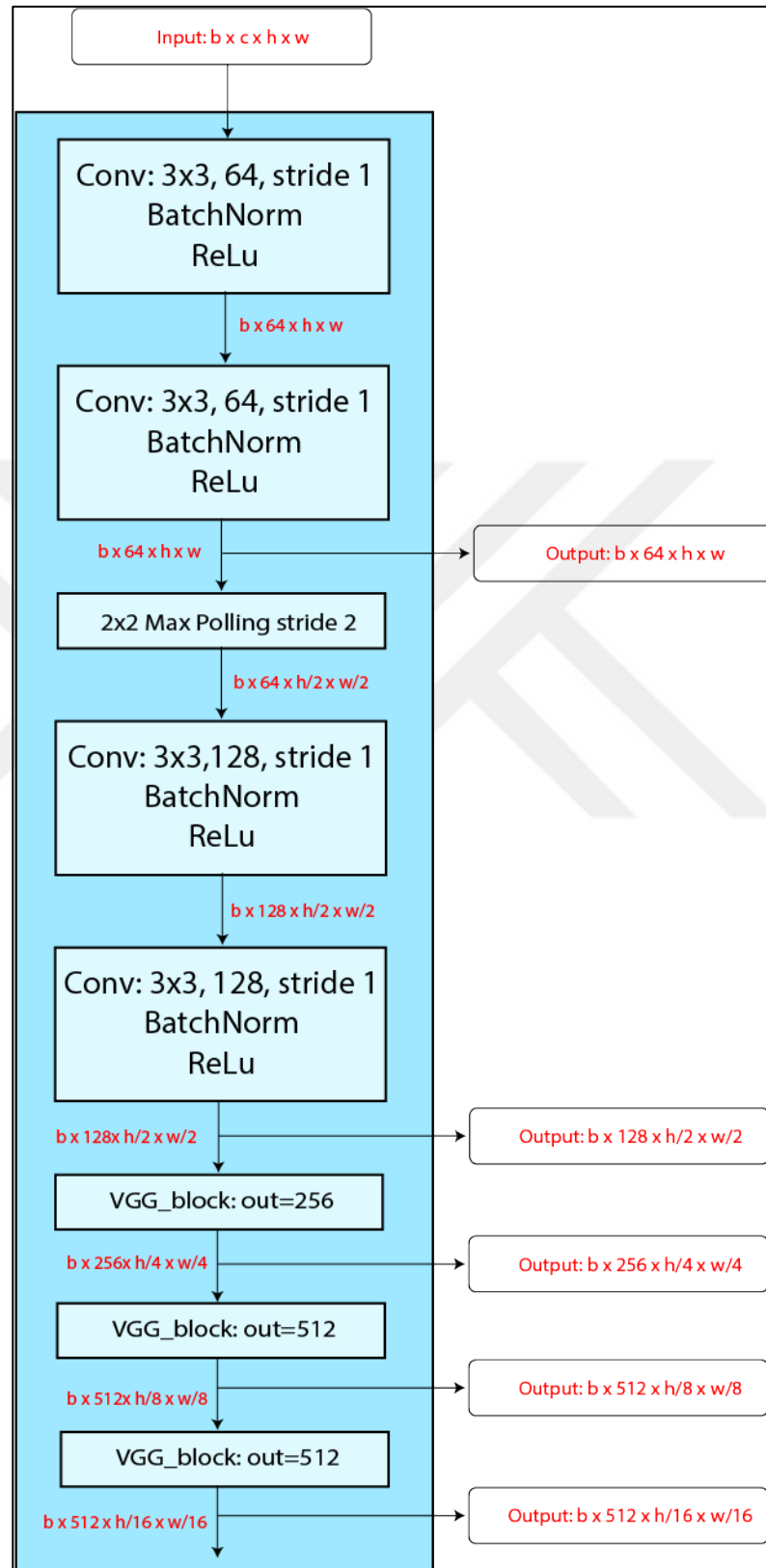


Figure B.1: VGG-16 Network Architecture.

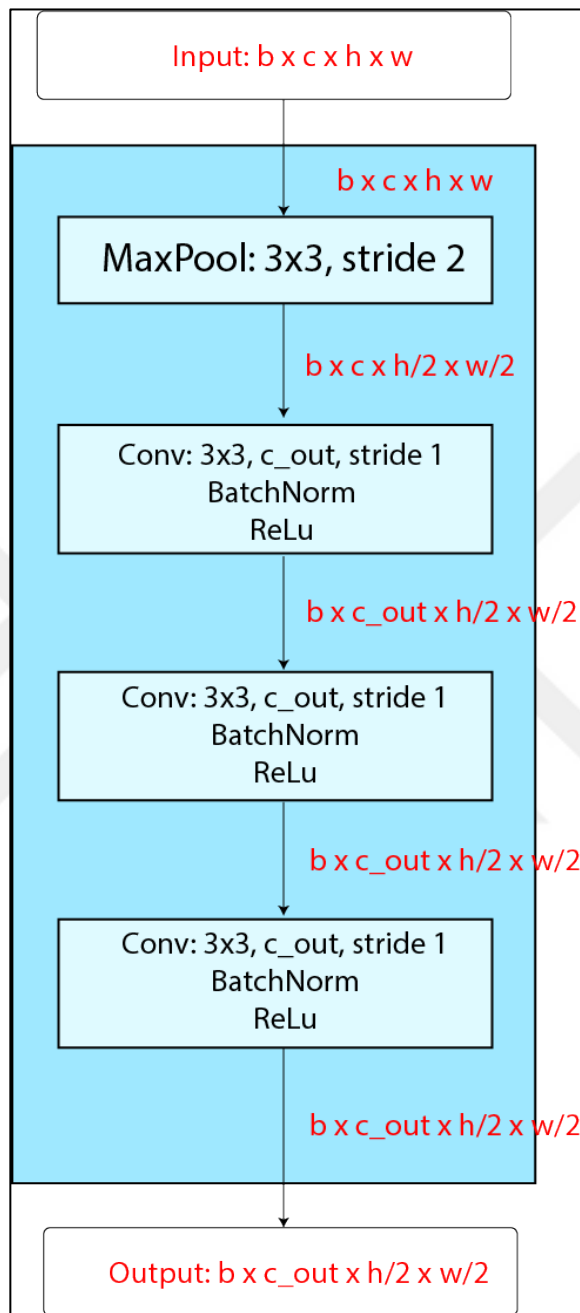


Figure B.2: VGG Block.