Ph.D. DISSERTATION

Congestion-scale-aware Design of Network Structure and Training Strategy for Crowd Density Estimation

군중 밀도 예측을 위한 네트워크 구조와 훈련방법의 혼잡도 및 크기 인식 설계

BY

Jiyeoup Jeong

February 2022

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

Congestion-scale-aware Design of Network Structure and Training Strategy for Crowd Density Estimation

군중 밀도 예측을 위한 네트워크 구조와 훈련방법의 혼잡도 및 크기 인식 설계

BY

Jiyeoup Jeong

February 2022

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Congestion-scale-aware Design of Network Structure and Training Strategy for Crowd Density Estimation

군중 밀도 예측을 위한 네트워크 구조와 훈련방법의 혼잡도 및 크기 인식 설계

지도교수 최 진 영

이 논문을 공학박사 학위논문으로 제출함

2021년 10월

서울대학교 대학원

전기 정보 공학부

정 지 엽

정지엽의 공학박사 학위논문을 인준함

2021년 12월

위 원 장	조 남 익
부위원장	최 진 영
위 원	고 형 석
위 원	정 교 민
위 원	최 종 원

Abstract

This dissertation presents novel deep learning-based crowd density estimation methods considering the crowd congestion and scale of people. Crowd density estimation is one of the important tasks for the intelligent surveillance system. Using the crowd density estimation, the region of interest for public security and safety can be easily indicated. It can also help advanced computer vision algorithms that are computationally expensive, such as pedestrian detection and tracking.

After the introduction of deep learning to the crowd density estimation, most researches follow the conventional scheme that uses a convolutional neural network to learn the network to estimate crowd density map with training images. The deep learning-based crowd density estimation researches can consist of two perspectives; network structure perspective and training strategy perspective. In general, researches of network structure perspective propose a novel network structure to extract features to represent crowd well. On the other hand, those of the training strategy perspective propose a novel training methodology or a loss function to improve the counting performance.

In this dissertation, I propose several works in both perspectives in deep learning-based crowd density estimation. In particular, I design the network models to be had rich crowd representation characteristics according to the crowd congestion and the scale of people. I propose two novel network structures: selective ensemble network and cascade residual dilated network. Also, I propose one novel loss function for the crowd density estimation: congestion-aware Bayesian loss.

First, I propose a selective ensemble deep network architecture for crowd density estimation. In contrast to existing deep network-based methods, the proposed method incorporates two sub-networks for local density estimation: one to learn sparse density regions and one to learn dense density regions. Locally estimated density maps from

the two sub-networks are selectively combined in an ensemble fashion using a gating network to estimate an initial crowd density map. The initial density map is refined as a high-resolution map, using another sub-network that draws on contextual information in the image. In training, a novel adaptive loss scheme is applied to resolve ambiguity in the crowded region. The proposed scheme improves both density map accuracy and counting accuracy by adjusting the weighting value between density loss and counting loss according to the degree of crowdness and training epochs.

Second, I propose a novel crowd density estimation architecture, which is composed of multiple dilated convolutional neural network blocks with different scales. The proposed architecture is motivated by an empirical analysis that small-scale dilated convolution well estimates the center area density of each person, whereas large-scale dilated convolution well estimates the periphery area density of a person. To estimate the crowd density map gradually from the center to the periphery of each person in a crowd, the multiple dilated CNN blocks are trained in cascading from the small dilated CNN block to the large one.

Third, I propose a novel congestion-aware Bayesian loss method that considers the person-scale and crowd-sparsity. Deep learning-based crowd density estimation can greatly improve the accuracy of crowd counting. Though a Bayesian loss method resolves the two problems of the need of a hand-crafted ground truth (GT) density and noisy annotations, counting accurately in high-congested scenes remains a challenging issue. In a crowd scene, people's appearances change according to the scale of each individual (*i.e.*, the person-scale). Also, the lower the sparsity of a local region (*i.e.*, the crowd-sparsity), the more difficult it is to estimate the crowd density. I estimate the person-scale based on scene geometry, and I then estimate the crowd-sparsity using the estimated person-scale. The estimated person-scale and crowd-sparsity are utilized in the novel congestion-aware Bayesian loss method to improve the supervising representation of the point annotations.

The effectiveness of the proposed density estimators is validated through comparative experiments with state-of-the-art methods on widely-used crowd counting benchmark datasets. The proposed methods are achieved superior performance to the state-of-the-art density estimators on diverse surveillance environments. In addition, for all proposed crowd density estimation methods, the efficiency of each component is verified through several ablation experiments.

keywords: crowd density estimation, crowd counting, scene understanding, visual surveillance

student number: 2014-21714

Contents

List of Tables

List of Figures

Chapter 1

Introduction

As a key component of the intelligent surveillance system, crowd counting has been extensively developed to provide security and safety for both people and infrastructure. Crowd counting has been utilized to detect abnormal surveillance situations and to maintain adequate crowd density for public safety. In particular, a situation to be aware of in visual surveillance is a high-congested situation. Such a situation could be a target for terrorism or for crowd control such as population decentralization. It has been applied to other computer vision-based tasks, such as cell counting [1], vehicle counting [2], or bio-statistical research [3] to boost up process performance and quality control. Usually, the crowd counting targets a high-congested situation like thousands of people are represented in one image. When a high-congested situation causes severe occlusions, however, a detection-based crowd counting algorithm often provides poor estimation for such a situation. Furthermore, unevenly distributed crowds, variability in pedestrian size, and diversity of surveillance environments make crowd counting more difficult.

In contrast to the conventional methods based on detection, a regression-based crowd counting algorithm has been introduced to resolve the aforementioned issues, which regresses the number of people with local features extracted from the image. Regression-based algorithms are mainly researched as following properties; the privacy-

preserving to counting target and the robustness for changing monitoring environment. In the regression-based crowd counting, a crowd density map is regressed from local features extracted from the input image. A density map is regressed rather than the count itself, crowds can be counted in any region of the image by integrating the density map. The regression-based approach can reliably count crowds in difficult situations such as high congestion or heavy occlusion because it uses the appearance patterns of crowds rather than individuals. A number of recent works train such density estimators to perform crowd counting in difficult surveillance situations with huge scale and various sparsity of the crowds.

As the recent introduction of deep learning, most of the crowd counting tasks adopt the common scheme that trains a convolutional neural network with a pair of images and their corresponding density maps. Datasets of the crowd counting generally provide images and locations of people in a pixel. Conventional methods make a ground truth (GT) density map by filtering the Gaussian kernel onto people's location. And if necessary, scene information such as geometric distortion or region-of-interest can be utilized to generate a more informative GT. For example, some studies generate GT by controlling the standard deviation of the Gaussian kernel to represent a scale of pedestrian [4], or by using a human-like shaped kernel [5].

Although deep learning-based crowd density estimation researches increase the level of understanding using properties of deep networks, there is still room for more improvement counting performance with utilizing representation of crowd feature. There are two key components for utilizing deep learning for a certain computer vision task; network structure and training details (*e.g.*, loss function or training order to train multiple networks). On one hand, with a deep inspection of deep networks for crowd density estimation, I propose two novel network structures for accurate crowd density estimation. First, I propose a selective ensemble deep network architecture incorporating two sub-networks for local density estimation: one to learn sparse density regions and one to learn dense density regions. Second, I propose a composed network including

multiple dilated convolutional neural network blocks with different scales of person. On the other hand, with an analysis of characteristics that make training of crowd density accurate, I propose one novel loss function for accurate crowd density estimation. I propose a novel congestion-aware Bayesian loss method that considers the scale of a person (*i.e.*, person-scale) and the sparsity of local crowd (*i.e.*, crowd-sparsity).

Chapter 2

Related Works

As known as crowd counting, crowd density estimation based on images is one of the fundamental problems in the visual surveillance [6]. The estimated crowd density can help to profile the crowd flow and to detect the region of interest in a surveillance scene [1]. However, due to the large variance in the appearance of crowds, enormous occlusion patterns, and diverse illumination conditions, crowd counting is a challenging problem to be solved. Crowd counting methods can be roughly classified into two groups: detection-based and regression-based methods [6].

2.1 Detection-based Approaches

Detection-based methods directly identify each target of the count using appropriate pedestrian detectors. When highly congested crowds are formed, however, the appearance of individuals may not be preserved in images, which results in poor algorithmic estimation. To resolve such occlusion issues, some studies use other types of detection targeting, such as faces [7] or the head and shoulders [8]. Despite such efforts, if the surveillance environment is changed, missing or false detection might be caused. Detection-based methods also have unnecessary computational costs that may not require the exact location of people for the sole purpose of counting crowds. Khan *et*

al. [9] utilized the results of a conventional head detector to accurately count people by warping the image patch according to the scale of a person. Also, Khan *et al*. [10] proposed the method using multi-scale fusion module for conventional pedestrian detection [11].

2.2 Regression-based Approaches

As rigorously reviewed in [6, 12], crowd counting has commonly employed regression-based methods, which obtain the number of people by estimating and integrating a pre-defined crowd density map. Regression-based methods are more frequently utilized because they perform better than detection-based methods in high-congestion situations or when there is severe occlusion. When regression-based methods were initially proposed, many studies performed mapping of low-level features directly to the size of the crowd [1, 13, 14]. However, these methods that use direct mapping to the count lack spatial information of the crowd so they cannot determine where counting errors occur.

To resolve the limitation of losing spatial information, Lempitsky and Zisserman [15] proposed a method that conducts a mapping of local features to an intermediate density map. The density-map regression method has since become the mainstream of crowd counting, enabling the counting of individuals in any region by numerical integration over a density map. Pham *et al*. [16] proposed a non-linear mapping method using the random forest algorithm. Wang and Zou [17] decreased ineffective computational complexity by using subspace learning in the mapping of a density map.

2.3 Deep learning-based Approaches

Recently, deep learning-based methods [2, 4, 5, 18–22] have significantly improved counting accuracy using rich feature representation of deep features by comparison of conventional methods that use hand-crafted features.

2.3.1 Network Structure Perspective

There have been several works for robustly coping with more complicated scenes via a network structure combining various types of features. A convolutional neural network (CNN) was first applied to estimate the crowd density map by Zhang *et al*. [4]. Motivated by this pioneering work, many studies have been conducted in which a deep network has effectively learned given pairings of an image and its density map. Zhang *et al*. [5] proposed a multi-column network to estimate crowds with varying scales trained in each network column.

There have been several works that use multiple networks to address the multi-scale problem. Sam *et al*. [19] proposed to switch a neural network that classifies image patches according to scale and estimates the crowd density separately. Daniel *et al*. [2] proposed a hydra-shaped network structure that resizes the image patch to several scales and combines the estimated crowd density. Sindagi *et al*. [23] introduced an auxiliary classifier to extract the scale features of an image patch then fused it to a conventional density estimator. Jeong *et al*. [24] constructed multiple network branches according to the sparsity of a local crowds and adopted multi-level refinement network to improve the density estimation accuracy. Hossain *et al*. [25] defined 'scale' as the number of people in a local region and proposed a density estimator using 'scale' as an additional feature. Khan *et al*. [26] performed small- and large-scaled crowd density estimation successively to improve the accuracy of density estimation. Shang *et al*. [18] employed spatio-temporal features as contextual information to estimate the crowd density map of an image. A spatio-temporal model for crowd density estimation that utilizes the temporal correlation between neighboring frames was proposed by Xiong *et al*. [20]. Li *et al*. [21] adopts dilated CNN to robustly estimate crowd density even in the highly congested scenes. Shen *et al*. [22] proposed a novel training method by dividing the image into grids and defining the adversarial loss between the crowd density of the whole image and the that of grid images. Ranjan *et al*. [27] improved counting performance by integrating feature maps from the estimation

on a low resolution to that on high resolution. Cao *et al*. [28] proposed multi-scale aggregation method by aggregating convolution kernels of various sizes. As similar to the proposed method, the aforementioned methods utilize the multiple networks to improves counting performance.

Several approaches use novel network modules that differ from basic components such as convolution and pooling layers. Li *et al*. [21] used a dilated convolution to effectively extract features of a large field of view. Liu *et al*. [29] used spatial pyramid pooling to adaptively encode the scale as contextual information. Ma *et al*. [30] used human scale quantization at multiple scale levels and trained additional networks to represent scale of a person with a combination of pre-defined scales.

2.3.2 Training Strategy Perspective

There have been several works that address the limitation of problem setting in crowd density estimation. They tackle (1) the objective function for training and (2) the generation process of the ground truth (GT) density map.

First, several studies pointed out the limitation of the L2 loss between the GT density map and the estimated density map that there is a discrepancy that high-quality representation of density map does not lead to accurate counting. Liu *et al*. [31] utilized the fact that the number of people in a sub-region will always greater than that in the region inside, and proposed what they described as a ranking loss. Shen *et al*. [22] proposed a cross-scale consistency pursuit loss method by using the fact that there is a relationship in which the entire density map is the sum of the partial density maps. Cheng *et al*. [32] designed a spatial awareness loss method to generate a loss when the number of people changes, not when the distribution of people is changed.

In contrast, some studies have tackled the limitations of the definition of the GT density map. Zhao *et al*. [33] utilized auxiliary tasks, such as the estimation of depth, along with the density map to improve the performance. Wan and Chan [34] proposed an adaptive density map generation process that generates learnable density map rep-

resentations to create sub-optimal density maps. Some works employed segmentation maps [35], the number of people as trainable sources [36], or pedestrian detection results [37]. In particular, Ma *et al*. [38] successfully resolved issues for the training objective and the need for the generation process of a GT density map. They proposed a novel loss (*i.e.*, the Bayesian loss) using the probability of indicating each pixel is included in each point annotation or background.

Chapter 3

Selective Ensemble Network for Accurate Crowd Density Estimation

3.1 Overview

For intelligent surveillance systems, counting people in crowded areas is an important task, as such places are the most common surveillance targets. People counting can be used in various applications, such as abnormal behavior detection, retail analysis, and security. One direct solution to count people is to detect all pedestrians and count them [39, 40]. However, in cases of low resolution or heavy occlusion, it is extremely difficult to automatically detect all the people in a crowded scene. To overcome this difficulty, an approach based on crowd density estimation has been actively studied [1, 2, 5, 14–16, 41–45], in which crowd density is defined as the number of people per unit area. However, crowd density estimation is challenging problem because of varying characteristics of pedestrians like shape, size, and height/width ratio depending on camera installation settings. Other factors that make crowd density estimation difficult include occlusions, background clutter, and non-uniform distribution of people.

As reviewed in [6, 12], people counting has commonly employed regression-based methods, which estimate number of people [1, 13] or crowd density [15, 16, 41] by

regression from extracted image features. These methods are robust to occlusion because they learn regression from image features rather than by counting people one by one. The purpose of learning in regression-based methods is to learn a regression function that estimates density values for each pixel in the extracted features, namely crowd density map. Recently, deep learning-based methods [2, 4, 5, 20, 42] have improved counting accuracy using rich feature representation of deep features by comparison of conventional methods that use hand-crafted features. Zhang *et al.* [5] have proposed a multi-column neural network to extract features at multiple scales for image-wise density map estimation. Similarly, a scale-aware network model called Hydra CNN was proposed in [2] to resolve the scale issue. Shang *et al.* [18] have estimated the crowd density map for a whole image through contextual information using recurrent networks. Switching convolutional networks [42] has been proposed as a means of learning crowd density map regression in various crowd environments. Xiong *et al.* [20] proposed a spatio-temporal model for crowd density estimation that utilizes temporal correlation between neighboring frames.

These CNN-based methods generally train the network with L_2 loss between the estimated density map and the ground truth density map. However, L_2 loss training in the density map is informed mainly by region with large density values, resulting in biased with that values. Also, most of these networks incorporate a feature abstraction step, such as a pooling layer. Although the feature abstraction step can improve counting accuracy by excluding ambiguous features, the resolution of the estimated density map is inevitably reduced, leading to inaccurate density estimation that undermines the density map accuracy of people counting.

To address these limitations, we propose a novel deep network structure and a training method to improve both counting accuracy and density map accuracy. The design of the proposed network is based on three key ideas: the incorporation of a new form of loss for more accurate density map estimation, the use of different sub-networks depending on crowd density, and the cascading of a refinement sub-network to take

account of contextual information. The first idea is to use a counting loss in addition to density map loss. To successfully leverage counting loss, we propose weighting scheme in the loss function that is adjusted according to degree of crowdness and training epoch. Second, in order to apply counting loss differently according to degree of crowdness, we propose two sub-networks: a sparse sub-network (for sparse crowd regions) and a dense sub-network (for dense crowd regions). These two sub-networks build an ensemble by selectively utilizing one of the two, based on the output of a gating sub-network. Finally, to increase the resolution of the estimated density map, we propose a refining sub-network to improve both density map and counting accuracy.

Experiments conducted on various datasets show that the proposed method achieves state-of-the-art performance in counting accuracy with reasonable localization quality.

3.2 Combining Patch-based and Image-based Approaches

Like Fig. 3.1, the CNN-based crowd density estimation methods can be categorized by two directions: patch-wise regression and image-wise regression.

The patch-wise regression methods [46–49] estimate the crowd density maps of local patches, which are aggregated to the final crowd density map of the entire image. Because a lot of training patches can be obtained from one training image, the patch-wise regression methods are able to describe the detail of the crowd density. However, because the patch-wise methods concentrate on only the local patches, the semantic context, such as impassable regions and occluded regions, cannot be considered. The insufficient context information causes much noisy density on the background region after the aggregation of the patch-wise crowd density map.

The image-wise regression methods [50,51] estimate the crowd density map directly from the frame image. Contrary to the patch-wise regression methods, the semantic context is easy to be trained because the results from the CNN would be different for the various position in the image. However, when the entire image becomes an input

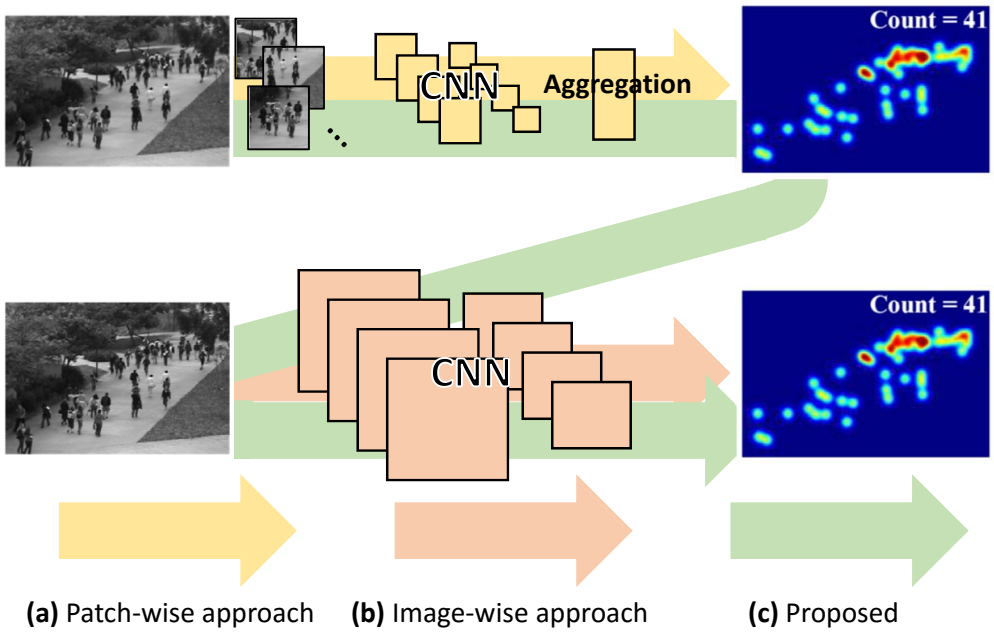


Figure 3.1: **Comparison of Image-based and Patch-based Crowd Density Estimation.** the arrow (a) represents a patch-wise approach, arrow (b) represents a image-wise approach, and arrow (c) represents our algorithm

of CNN, the details of the crowd density map can be suppressed by the insufficient number of training data. The scale variance of crowds also cannot be resolved because of a fixed receptive field of networks in these methods. Even though some works [51] tried to solve the scale variance problem by integrating multiple networks of various receptive fields, only the discrete size of receptive fields could be considered, and the number of the weights associated with the entire network increases dramatically.

In this sub-chapter, we propose a novel deep architecture for CDE which conserves the detail of the crowd density map, while minimizing the noisy density by considering the semantic context. The proposed CNN architecture consists of two sub-networks: Local Network (L-Net) which estimates local crowd density maps of patches and Global Network (G-Net) to estimate the final crowd density map of the input image. L-Net estimates local crowd density maps of the patches extracted from the entire image so that the detail of the crowd density map is preserved. G-Net integrates the detail patch-wise crowd density map by considering the semantic context by the recursive network. In addition, due to the recursive structure of G-Net, the model complexity could be reduced even with a deeper network structure, while the over-fitting problem caused by many associated weights can be prevented. With several experiments, the proposed network shows the state-of-the-art performance as well as represents a sophisticated crowd density map. The main contributions of this sub-chapter are summarized as below:

- We propose a novel network architecture Local-Global Cascade Network to estimate accurate crowd density map with an accurate configuration of crowd density map.
- By utilizing recursive network structure in G-Net, we can reduce model complexity as well as can employ various receptive fields using outputs from intermediate layers.
- We get better results over the prior state-of-the-art works and confirm the effec-

tiveness of the proposed method.

3.2.1 Local-Global Cascade Network

In general, crowd density estimation is formulated with mapping between images and its pixel-wise density map like

$$F : X \rightarrow D, \tag{3.1}$$

where X is the feature extracted from the image and D is the corresponding density map. Because density values of the crowd are hard to define, we use density map convoluting with the representative crowd density mask to people locations of images for the ground truth. Representative density mask we used is depicted in Fig. 3.3 (a) and convoluted people location map is like Fig. 3.3 (b). By re-sizing the mask according to the perspective map as shown in the lower right of Fig. 3.3 (b), the resulting ground truth crowd density map is like Fig. 3.3 (c). We conduct CDE in the image considering two objectives; the counting accuracy and the accuracy of crowd density map configuration. Our proposed CNN model is designed cascade-connected two sub-networks like Fig. 3.2 to achieve the aforementioned two objectives. At first, Local Network (L-Net) estimates partial crowd density of images using grid-cropped patches, then after patch aggregation, Global Network (G-Let) estimates final crowd density map using initial crowd density map. Though L-Net has chances to falsely estimate crowd density of patches' such as positive density values on background, G-Net can refine these flaws using semantic reasoning.

Overall Framework

The main problem in Eq. 3.1 is that the image has various sizes of crowds. So, direct mapping between an image and its crowd density map is not available. To overcome this imbalance issue, we design a network with the collaboration of two sub-networks. At first, an input image is cropped with the same size patches and passes through L-Net

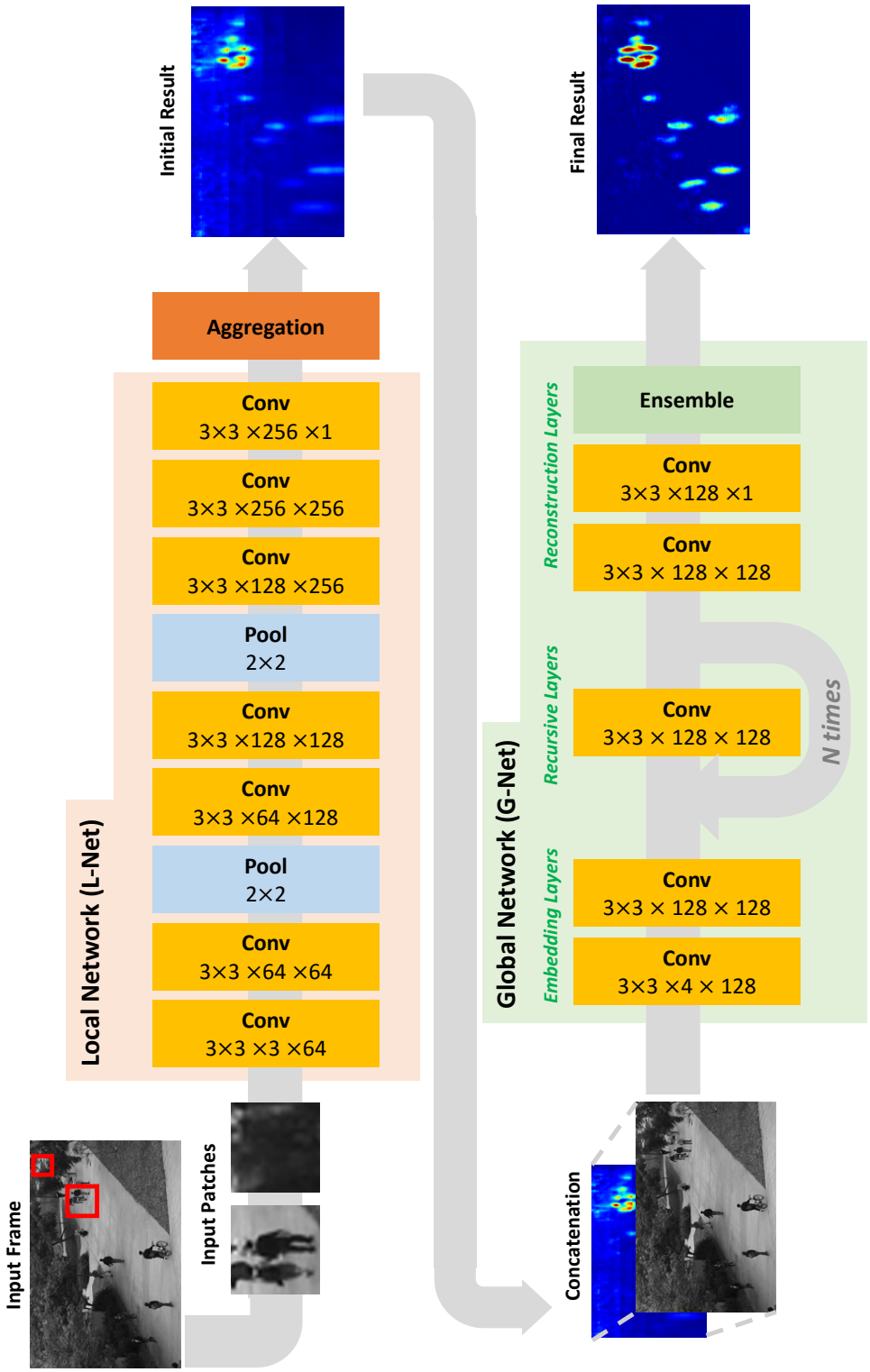


Figure 3.2: **Framework of the proposed algorithm** Upper part of the figure is L-Net and lower one is G-Net, respectively

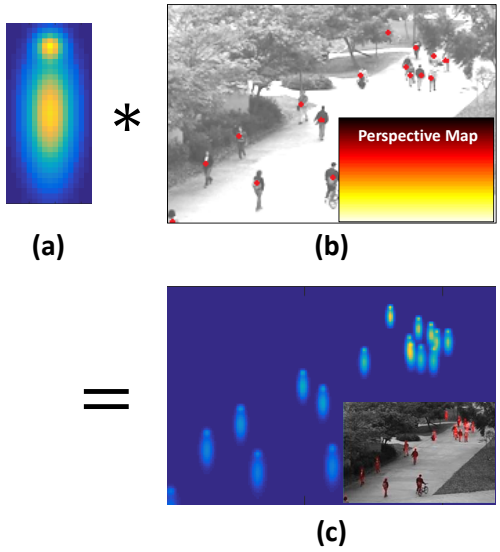


Figure 3.3: **Generation process of the ground truth crowd density map** The ground truth crowd density map (c) is generated by convolution between representative density map of a person (a) and its locations (b)

to estimate crowd density locally. The resulting crowd density maps of the patches are aggregated to a crowd density map of the original input image through the process of averaging the overlapping regions. We then refine the initial result in G-Net to amplify true values of crowd density and to suppress false estimated ones.

Local Network (L-Net)

A perspective map of the specific scene is created using two bounding boxes of certain two-person as shown in the lower right of Fig. 3.3 (b). Motivated by [13, 48], we interpolate the width and height values of two bounding boxes to roughly approximate the size of people bounding boxes in the entire region of the image. A square root value of the perspective map is used in the generation process of the ground truth crowd density map and cropping step to make image patches. Input patches in L-Net are created by grid-wise cropping input image using the perspective map with overlapping

neighboring patches. In our work, we use the size of the input in L-Net as 100×100. In more detail, the patch size on the location where the representative crowd density mask is determined is 100×100 and that on the other location is re-sized by the ratio of values of the perspective map. As in the upper part of Fig. 3.2, the base network structure of L-Net is a revised version of a pre-trained model trained on image-classification task [52] to express the crowd in highly expressive power. We revise the VGG-16 model [52] by adding one convolutional layer whose output has one channel at the end of *Conv 3-3*. That is, additional layer has a size of $3 \times 3 \times 256 \times 1$. In optimization of L-Net, we use loss function as L_2 loss between ground truth and its estimated crowd density as follows:

$$L_{local}(\theta) = \frac{1}{Z} \sum_{p \in P} \left\| d_p - \hat{d}_p \right\|^2 + \lambda \|\theta\|^2 \tag{3.2}$$

where θ is network parameter, Z is normalization value, P is set of pixels in the patch, and both d_p and $\hat{d}_p$ are true and estimated pth pixel value of crowd density map. The optimizer we used in L-Net is *RMSProp* optimizer [53]. To prevent over-fitting of L-Net, we make up a mini-batch with the same number of positive and negative samples. Here, a negative sample means a sample with few people in the patch. The resulting crowd density maps of the patch are aggregated with averaging to be the initial crowd density map to be the same size of the input image. The aggregation process is nothing but summing all patch crowd density results at their location and dividing the summed result by the number of overlapping times.

Global Network (G-Net)

In order to refine the initial crowd density map after L-Net, we use another sub-network G-Net whose input is concatenated image with the initial crowd density map and the original input image. Basically, G-Net should have a larger size receptive field than one of L-Net because we should correct the initial crowd density map considering a large portion of the image. For this reason, we need a deep network structure in G-Net.

According to [54], we can design a deep network with low model complexity using recursive network structure. Motivated by [55], we also use intermediate outputs of recursive layers as another feature to reconstruct the final crowd density map of the image so that the output of G-Net considers various receptive fields even in a single network. As in the lower part of Fig. 3.2, G-Net consists of three groups of layers like [54]; embedding layers, recursive layers, and reconstruction layers. All group of layers is configured with two convolution layers except the final reconstruction layer. The final reconstruction layer has a kernel whose size is $1 \times 1 \times (recursive\ time + 1) \times 1$ in order to combine intermediate features of recursive layers and features at the end of the recursive layer. The input channel size of this layer is $(recursive\ time + 1)$ because it also uses the initial feature after the embedding layers. In other words, the result from the final reconstruction layer is equal to the weighted sum of the results of each recursive layer, and we call this result an ensemble result. G-Net uses L_2 loss between the ground truth and its estimated crowd density map in both the results in each recursive layer and the ensemble result as follows:

$$L_{\text{inter}}(\theta) = \frac{1}{R \times Z} \sum_{r=1}^{R} \sum_{p \in P} \left\| d_p - \hat{d}_p \right\|^2 \tag{3.3}$$

$$L_{ensemble}(\theta) = \frac{1}{Z} \sum_{p \in P} \left\| d_p - \sum_{r=1}^{R} w_r \times \hat{d}_p \right\|^2 \tag{3.4}$$

$$L_{global}(\theta) = \alpha L_{\text{inter}}(\theta) + (1 - \alpha) L_{ensemble}(\theta) + \lambda \|\theta\|^2 \tag{3.5}$$

where R is recursion times, Z is normalization value, P are set of pixels in the input image, and both d_p and $\hat{d}_p$ are true and estimated pth pixel value of crowd density map. w_r is the weight of rth recursive layer and α represents the weight between the two losses. Eq. 3.3 is the loss that makes the crowd density map reconstructed from each intermediate layer equal to the final crowd density map. By ensuring that both the ensemble crowd density map and the reconstructed crowd density map are equal to the final crowd density map in the total loss like Eq. 3.5, the training process of G-Net is done properly. The optimizer we used in G-Net is *Adam* optimizer [56].

Implementation Detail

The detailed network configuration and its filter size are all illustrated in Fig. 3.2. In the generation process of the patches in L-Net, we define the positive patch as the patch containing more than 0.99 crowd density i.e. there is almost one person in the patch, and the negative patch as the patch containing less than 0.01 crowd density i.e. there are few people in the patch. In L-Net, the weights of all layers are initialized by pre-trained one except the additional layer. The weights of the kernel in the additional layer are initialized by truncated normal values with 0.1 standard deviations. The input patches of L-Net are re-sized to 100×100 and the output patches are up-sampled 4 times using bicubic interpolation. It is because the VGG-16 model [52] up to *Conv 3-3* has two pooling layers, resulting in 4 times being smaller than the input. In the training process of L-Net, we iterate 1.2×10^6 images with a mini-batch consisting of 16 patch images. Iteration is progressed in chronological order at first 6×10^6 iteration and after 6×10^6 iteration is progressed randomly. The learning rate of *RMSProp* optimizer is planned to be reduced 10^{-1} times from 10^{-7} to 10^{-8} by dividing the entire learning process into two parts. In the patch aggregation process after L-Net, we use an overlap ratio between patches as 0.3 i.e. 0.3 portions of neighboring patches are overlapped. After L-Net is trained, we start to train G-Net. We concatenate the initial crowd density map with the input image at the third channel and there is no channel normalization to make it the input of G-Net. All weights of kernels in G-Net are initialized by *Xavier initialization* [57] except recursive layers. The weights of kernels in recursive layers are initialized by zeros. In the training process of G-Net, we iterate 8×10^4 times with a mini-batch consisting of 5 images. Half the number of iteration is progressed in chronological order and the other half is done randomly. The learning rate of *Adam* optimizer is planned to be reduced 10^{-1} times from 10^{-4} to 10^{-6} by dividing the entire learning process into three parts and momentum value is set by 0. The α in Eq. 3.5 is firstly set by 0.5 and planned to be decreased by 0.1 depending on the learning rate schedule. We implement all of our proposed network with *TensorFlow* library [58].

3.2.2 Experiments

We use the public dataset UCSD [13] to evaluate performance compared to the state-of-the-art algorithms. UCSD dataset is a video taken on a university, which composed 2000 frame images including training frames whose frame numbers are [601:1400] and test frames whose frame numbers are [1:600, 1401:2000]. It has people in the range of 12-45 passing by the sidewalk and In the first study that provided datasets [13] provides region-of-interest (ROI). In order to compare counting accuracy between our work to the state-of-the-art algorithms, we use commonly used evaluation metrics; mean absolute error (MAE) and mean squared error (MSE). Two evaluation metrics are as follows:

$$MAE = \sum_{i=1}^{N} \frac{|y_i - \hat{y}_i|}{N} \tag{3.6}$$

$$MSE = \sum_{i=1}^{N} \frac{(y_i - \hat{y}_i)^2}{N} \tag{3.7}$$

where N is the number of frame images and y_i and $\hat{y}_i$ are true and estimated number of people in the ith frame image. Both metrics represent the quantitative value of how exactly the number of people is precisely estimated so we do not know the crowd density maps are well estimated by these metrics. Also, other state-of-the-art algorithms provide a few results as a type of crowd density map, we cannot compare the accuracy of crowd density map configuration quantitatively. So in order to verify the accuracy of crowd density map configuration, we conduct qualitative comparison experiments in the proposed model, which compares the initial crowd density map at the end of L-Net to the final crowd density map at the end of G-Net. We also show the efficiency of recursive network structure in G-Net by ablation experiments.

Counting Performance

To evaluate the counting performance, we compare our algorithm to several state-of-the-art algorithms with MAE and MSE metrics. Counting performance is summarized as

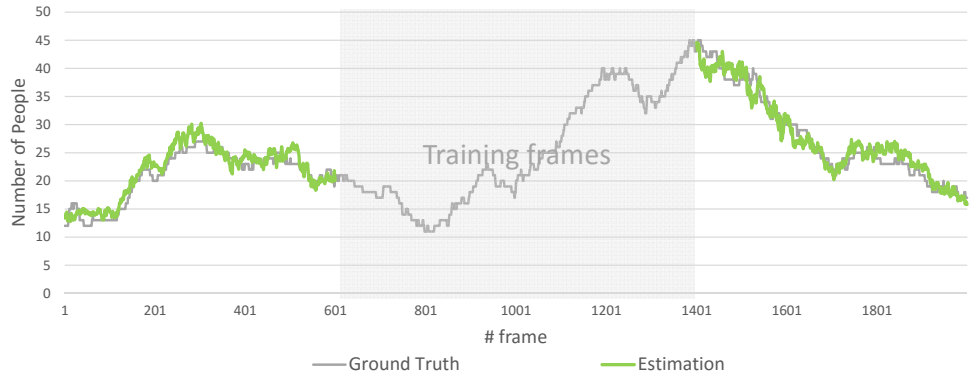


Figure 3.4: **Experimental Results of UCSD dataset** Each graph represents the predicted number of people and the actual number of people by the frame number

Figure 3.5: **Performance comparison graph based on recursion time**

Table 3.1: **Counting performance comparison**

	MAE	MSE
[59]	2.16	7.45
[1]	2.25	7.82
[13]	2.24	7.97
[14]	2.07	6.86
[48]	1.60	3.31
[51]	**1.07**	**1.35**
Ours (After L-Net)	2.79	14.68
Ours (After G-Net)	<u>1.47</u>	<u>3.24</u>

Table. 3.1 and the result graph for detail is shown in Fig. 3.4 Ours got the second-best performance in both MAE and MSE metrics. Among CNN-based algorithms, ours outperform the patch-wise approach [48]. It is because we consider various receptive fields in a single network structure different from [48] so that the proposed network could learn crowds of various sizes. However, ours got lower performance compared to the image-wise approach [51]. We think that it is because ours uses consecutive sizes of receptive fields from 2 to 10-pixel difference while [51] uses artificially-defined receptive fields through experiments. These artificially-defined receptive fields might improve the counting performance, though, they can lose practical applicability.

Table. 3.1 also shows the performance between L-Net and G-Net. Because L-Net is similarly designed patch-wise approaches for CDE, we could verify what is the difference ours to the patch-wise crowd density estimation. The result of L-Net is improved by G-Net in a large margin. In Fig. 3.6, we confirmed where the performance improvement comes from into two aspects. Firstly, false negatively estimated crowd density like lower density in crowded areas is re-activated through G-Net. This fact shows that G-Net improves performance by positive feedback to the crowded area.

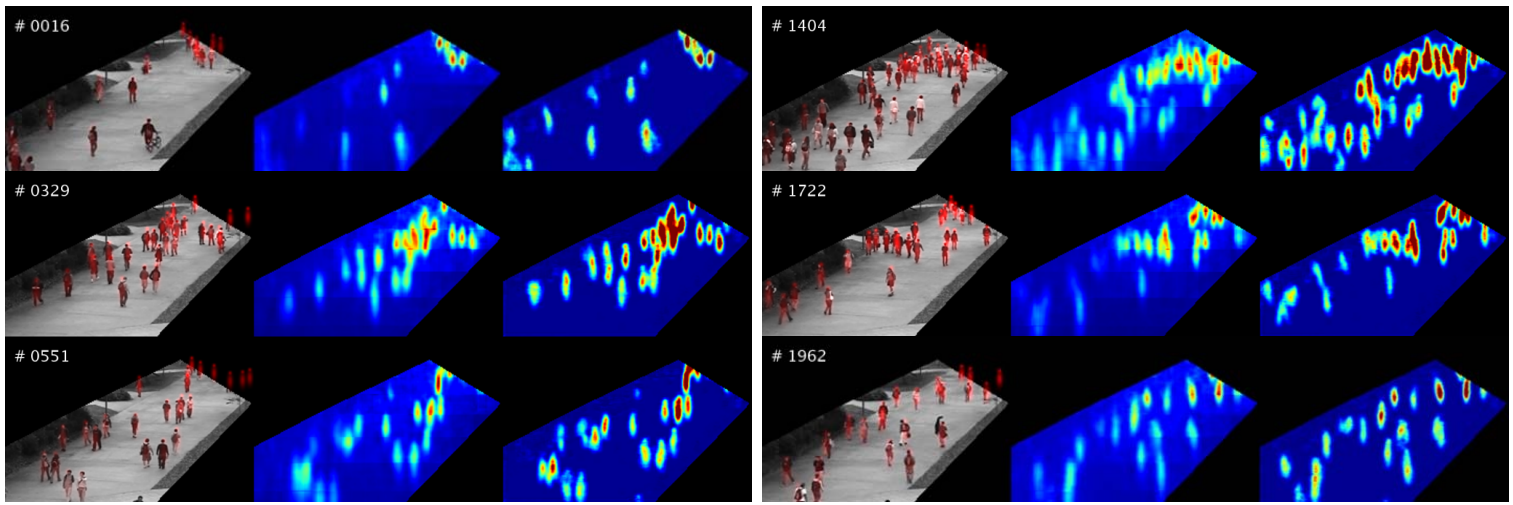


Figure 3.6: **Qualitative comparison of L-Net and G-Net** The left side of each column represents the input image with the ground truth crowd density map, the middle is the L-Net result, and the right is the G-Net result

Second, false positively estimated crowd density like density in background area is suppressed through G-Net. This shows that G-Net has improved performance by negative feedback in the background. As a result, the final crowd density map is more clearly represented after G-Net than the initial crowd density map. In short, mispredicted crowd density can be corrected using semantic reasoning *i.e.* large receptive fields through the whole network.

Analysis on Network Structure

In order to verify the efficiency of the proposed network structure of G-Net, we conduct some ablation experiments. At first, we test the correlation between the recursion times and the performance. We test our algorithm on 3, 5, 7 recursion times to see the difference in counting performance. As Fig. 5.5, we show that performance is saturated when the recursion time exceeds 5 times. It means that the representation of the crowd is enough to express the crowd density of the specific scene. One important thing is that there are no more parameters when the recursion time increase because all the recursive layers shares parameters. However, the more recursion time, the more training time required.

We also conduct experiments on our algorithm compared to the revised version without recursive network structure. Unfortunately, in the cases of 5 and 7 recursive times, the non-recursive network structure fails to learn because of exponentially many parameters *i.e.* the model complexity is high. This shows that recursive structure has the benefit that it can be a deeper network without increasing model complexity. Also, in order to balance model complexity and the number of receptive fields, the recursive network can be an adequate network structure.

3.2.3 Summary

In this chapter, we proposed Local-Global Cascade Network for CDE. We separated the crowd density estimation task into two sub-tasks as the estimation of local crowd

density with the image patches and the refinement initial crowd density map using the semantic cue. As a result, we could improve counting accuracy. Through several experiments, we confirmed that the refining process of the initial crowd density map is suitably implemented with recursive structure in CNN. Also, at the end of this sub-chapter, we discussed that there is a large room for improving actual performance through additional experiments such as defining the size of receptive fields, and also we discussed that the more efficient structure than our work can be proposed in the future using the end-to-end connection.

3.3 Selective Ensemble Network with Adjustable Counting Loss (SEN-ACL)

3.3.1 Overall Scheme

The proposed network architecture is depicted in Figure 3.7, which consists of four sub-networks: Gating Network (G-Net), Sparse Network (S-Net), Dense Network (D-Net), and Refinement Network (R-Net). Given image patches extracted from an input image, G-Net evaluates the degree of crowdness and determines which network, S-Net or D-Net, is appropriate to estimate the density of the given local patch. S-Net is trained for sparse patches, whereas D-Net is for dense patches. After estimating a density map for each patch, we aggregate every local patch density map to get an initial density map for the whole input image. The aggregation step is performed by element-wise addition and averaging overlapping regions. Then, an initial density map is refined by R-Net. The input of R-Net is constructed by concatenating the input image, the perspective map, and the initial density map. R-Net refines the initial crowd density map into a high resolution map considering contextual information of the image.

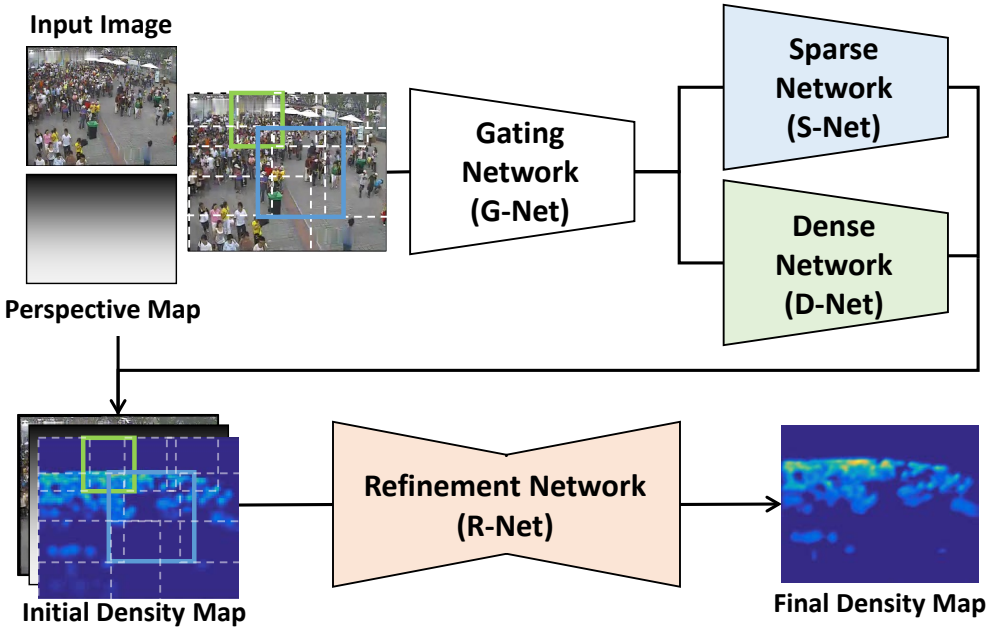


Figure 3.7: **The overall scheme of the proposed Selective Ensemble Network (SEN).** Dashed whited box represent extracted patches of the input image and the green (blue) box means a patch classified as *dense (sparse).*

3.3.2 Data Description

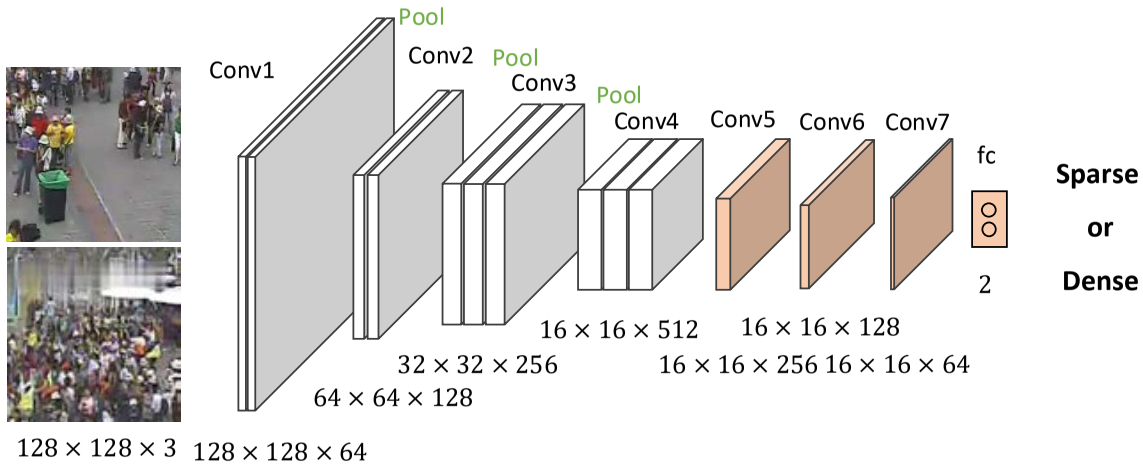
The crowd density map ground truth $D \in \mathbb{R}^2$ corresponding to an input image I represents the density of people appearing at each pixel as a real value. The number of people $C \in \mathbb{R}$ in a region D is obtained by integrating the density map. Our goal is to train the network so that the estimated crowd density map $\hat{D}$ (or the estimated number of people $\hat{C}$) is close to the ground truth crowd density map D (or the actual number of people C). As in [4], we design the ground truth for counting the whole body with different size depending on camera view angle. We apply Gaussian mask with different standard deviation so that partially appeared pedestrians also can be considered. The sizes of pedestrians in an image are depicted by the perspective map $P \in \mathbb{R}^2$. We generate P with linear assumption of the height of people as [4].

When extracting local patches from an image, we use P to decide the size of patch so that the size of a pedestrian in that patch should be consistent, which resolves the scale issue of pedestrians in various scenes. We extract local patches through sliding windows with overlapping to reduce mismatches in boundaries of the patches. The extracted patches are re-sized to a fixed size $W_p \times H_p$ for the input of the proposed network.

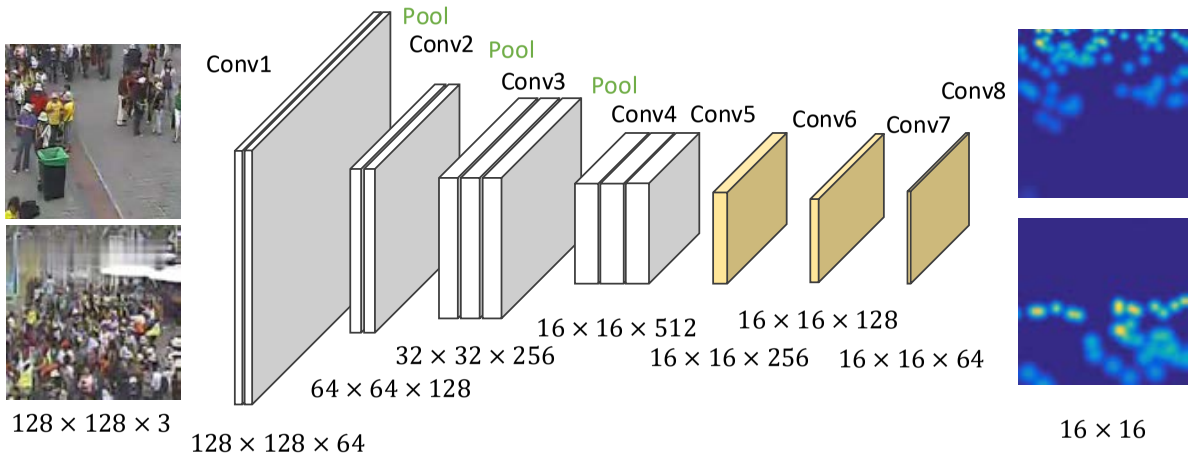
3.3.3 Gating Network

Gating Network (G-Net) decides which network (*sparse* or *dense* network) should be chosen for more accurate people counting results. For the n-th patch in N_p training patches, the ground truth label l_n for crowdness is defined as: $l_n = 1$, for sparse crowded image and $l_n = 2$ for dense crowded image. The desired output $y^{(n)}$ of G-Net is encoded by one-hot vector $(1, 0)$ or $(0, 1)$, respectively, and the soft-maxed output of G-Net is denoted by $\hat{y}^{(n)}$. The loss of G-Net L_G is defined as the cross-entropy loss as follows:

$$L_G = -\sum_{k}^{2} \sum_{n \in N_p} y_k^{(n)} \log(\hat{y}_k^{(n)}), \tag{3.8}$$



(a) Gating Network (G-Net)



(b) Sparse / Dense Network (S/D-Net)

Figure 3.8: The structure of **(a) Gating Network (G-Net)** and **(b) Sparse / Dense Network (S/D-Net)**

where m is the element index value of y. We design the network structure of G-Net with the help of the pre-trained network in order to use the rich feature representation of that network. In this work, we use VGG-16 [60] network structure up to *conv 4* layer and initialize with its pre-trained weights. Training patches are extracted in random positions.

The crowdness of the extracted patches are initially labeled by one of *dense* or *sparse* based on the number of people in those patches. In our case, when the maximum number of people in the training patches is N_M, the patch is labeled by *sparse* if the number of people in the patch is in $[0 : \frac{2}{3}N_M]$, and the patch is labeled by *dense* if it is in $[\frac{1}{3}N_M : N_M]$. Patches in the overlapped range $[\frac{1}{3}N_M : \frac{2}{3}N_M]$ are labeled randomly. While, these initial labels are not strictly correct although it has a tendency toward crowd density. Therefore, we train the networks with the initial crowdness labels first and then re-assign labels every epoch after S/D-Net are almost (80% in our work) trained. A crowdness label of the n-th patch l_n is re-assigned according to the counting accuracy of each network (S-Net and D-Net) as follows:

$$l_n = \arg\min_k \left| c_n - \sum_{x,y} \hat{d}_n^{(k)}(x,y) \right|. \tag{3.9}$$

c_n denotes the number of people and $d_n^{(k)}$ denotes the density map in S-Net ($k = 1$) or D-Net ($k = 2$) of n-th patch.

3.3.4 Sparse / Dense Network

Sparse / Dense Network (S/D-Net) regresses the density map of a given image patch. As shown in Figure 3.8-(b), the structure of S/D-Net is same as that of G-Net except last one convolution layer for density map regression. We design same architecture for both S/D-Net in order to train both networks only with the proposed loss scheme. The loss of the S/D-Net $L_{S/D}$ is defined by considering both the density map accuracy and the counting accuracy as follows:

$$L_{S/D} \quad = \quad \frac{1}{N_p}\sum_{n\in N_p}\left\|d_n-\hat{d}_n^k\right\|_2 \;+\; \lambda_k\frac{1}{N_p}\sum_{n\in N_p}(c_n-\sum_{x,y}\hat{d}_n^k(x,y))^2, \quad (3.10)$$

where N_p denotes the number of training patches. The first term of (3.10) indicates the accuracy of the estimated density map for n-th patch, which is called 'density loss'. And the second term indicates the accuracy of people counting estimated by the S-Net ($k=1$) or D-Net ($k=2$) for n-th patch, which is called 'counting loss'. When training S/D-Net, we set the weighting value λ_k between the two loss terms by considering the following two points. First, the counting loss interferes with training the density map because it can be minimized without accurate estimation of density map by canceling out the underly-counted and overly-counted region in integration of the density map. To lessen this interference, the weighting factor λ_k of the counting loss starts with a small value in the training phase before converging of density loss. As the training progresses, we increase the weighting factor λ_k of the counting loss gradually. Second, in the high crowded region, the features of pedestrian becomes unclear because of severe occlusions. In this region, the counting loss becomes more influential to counting accuracy than the density loss due to unclear features. Thus, D-Net uses a larger increment of λ_k than S-Net for each epoch.

At first training phase (less than 40% epochs in our work), both of S/D-Nets are trained using all training patches regardless of the degree of crowdness. This is because only using the dense crowd patches for D-Net is not enough to learn the appearance features of pedestrians due to severe occlusions. Before training S/D-Nets, the ground truth crowdness label l_n for n-th patch is reassigned with the output of G-Net as

$$l_n = \arg\max_k \hat{y}_k^{(n)}. \quad (3.11)$$

Note that l_n is re-assigned alternately by (3.9) and (3.11). That is, l_n is re-assigned on the basis of the outputs of S/D-Nets for training of G-Net, and l_n is changed by using the output of G-net for training of S/D-Nets. The entire training process is summarized in 1.

Algorithm 1: Training procedure of G/S/D-Net

Data: Training set with N_p training patches, N_E training epoches

Result: Parameters of G/S/D-Net

Initialize G/S/D-Net

for $i = 1...N_E$ **do**

  $\lambda_1 \leftarrow \lambda_1 + 0.01$ // λ for S-Net

  $\lambda_2 \leftarrow \lambda_2 + 0.1$ // λ for D-Net

  if $i < \frac{2}{5}N_E$ **then**

    Training S/D-Net with whole training patches with eq. (3.10)

  else if $i < \frac{4}{5}N_E$ **then**

    Training G/S/D-Net using initial labels with eqs. (3.8) and (3.10)

  else

    for $n = 1...N_p$ **do**

      Relabeling training patches with eq. (3.9)

    end

    Training G-Net with eq. (3.8)

    for $n = 1...N_p$ **do**

      Relabeling training patches with eq. (3.11)

    end

    Training S/D-Net with eq. (3.10)

  end

end

3.3.5 Refinement Network

Through Refinement Network (R-Net), we recover the resolution of the initial crowd density map with contextual information by regressing the undefined pixel values in the high resolution map, and also rectifies the errors occurred in the patch aggregation process. As shown in Figure 3.9, the input of R-Net is a tensor that concatenates initial crowd density map, perspective map, and the original image itself. Then, R-Net yields a refined crowd density map as the output. The structure of R-Net is motivated by the network structure of Convolutional Neural Pyramids (CNP) [44] and U-shaped Network (U-Net) [61]. The skip-connection for each scale [61] is utilized to preserve the details of the density map by learning residuals. Also, by using the pyramidal scheme [44] which learns the features with a wide range of scales efficiently, we combine the density maps of multiple scales into a refined density map containing contextual information.

R-Net consists of three modules as shown in Figure 3.9: feature extraction, mapping, and reconstruction. First, 64 channel features are extracted from the input tensor through a feature extraction module which consist of two 3×3 conv layers. The extracted feature tensor passes through two paths: one goes directly to the nonlinear mapping module composed of conv layers, and the other goes to the down-sampling (pooling) layers to form a half-sized feature tensor. This procedure is repeated d times consecutively, where d is called the scale pyramid level. The d-th level feature tensor has 2^{-d} size of the original feature tensor and goes directly to the nonlinear mapping module in d-th lavel. The nonlinear mapped feature tensor in d-th level is reconstructed into the lowest resolution density map. This density map passes the up-sampling layers and is pixel-wisely added with the density map reconstructed in the upper lavel. This reconstruction procedure is also repeated d-times and then the final crowd density map is produced. We use up-sampling layers as 3×3 deconv layers.

In the n-th image in the N_I training frame images, when the crowd density map and the number of people of the image are D_n, C_n, respectively. The output of R-Net

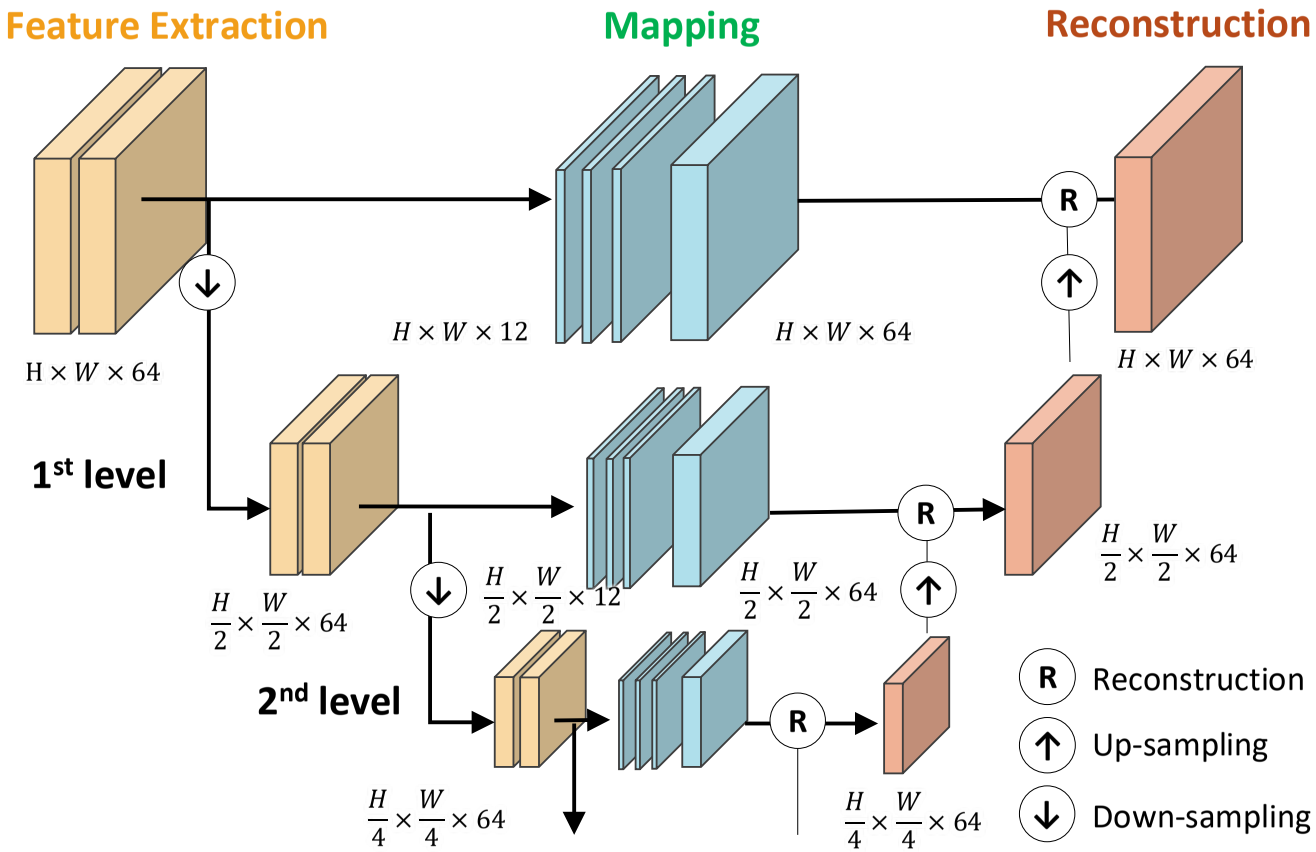


Figure 3.9: The structure of **Refinement Network (R-Net)**

Table 3.2: Dataset description. N_f : the number of frames, N_s: the number of scenes, R: the resolution, V: the number of people in the image, T_P: the total number of people.

Dataset	WorldExpo'10	UCSD	Mall
N_f	4.44 million	2000	2000
N_s	108	1	1
R	576×720	158×238	480×640
V	1∼253	11∼46	13∼53
T_p	199923	49885	62325

$\hat{D}_n$ denotes the refined crowd density map. The loss of R-Net L_R is given by

$$L_R = \frac{1}{N_I}\sum_{n\in N_I}\left\|D_n - \hat{D}_n\right\|_2 + \lambda\frac{1}{N_I}\sum_{n\in N_n}\left(C_n - \sum_{x,y}\hat{D}_n(x,y)\right)^2. \quad (3.12)$$

3.4 Experiments

3.4.1 Implementation Details

In the process of making the initial density map, the size of the patch is empirically determined to cover 3.5 × 3.5 (m^2) in the actual scene and we set the overlapping ratio of neighboring patches as 30% of width and height of the patch. All patches for training and test are resized to 128×128 by bicubic interpolation for the input of G/S/D-Net $W_p \times H_p$ so its corresponding density map size is 16×16. As described in Section 3.3.4, the initial λ_k in (3.10) is set to 0 for both S-Net and D-Net, and the increment of λ_k is set to 0.01 for S-Net and 0.1 for D-Net. For R-Net, we set the scale pyramid level d to 2 and λ in (3.12) starting at 0 with 0.05 increment for every epoch. We use *TensorFlow* library to implement the proposed network.

Table 3.3: Comparison errors (average MAE) for different settings on WorldExpo'10 [4]

BaseNet	G/S/D-Net	G/S/D-Net +ACL	G/S/D-Net +ACL+RP	**SEN-ACL**
17.0	13.1	11.0	10.3	**9.0**

3.4.2 Dataset and Evaluation Metrics

Three publicly available datasets were used to evaluate the proposed network's performance: WorldExpo'10 [4], UCSD [13], and Mall [1], as described in Table 5.2. Both Mean Absolute Error (MAE) and Mean Squared Error (MSE) between the estimated number of people and the ground truth were used to evaluate counting performance. These measure are widely used in crowd density estimation algorithms.

3.4.3 Self-evaluation on WorldExpo'10 dataset

An ablation study was performed to validate the effect of each element of the proposed method. This section compares one baseline and four variants of the proposed method. Each variant was derived by adding each element of the proposed method to the baseline as follows.

- BaseNet: Baseline is a patch-based regression model. It uses a single network for density map regression, and λ in (3.10) is fixed to 1.
- G/S/D-Net: Image patches were classified as *sparse* or *dense* using G-Net, and one counting result of the two networks (S/D-Net) was selected on the basis of G-Net results. The loss function of G-Net is shown in (3.8), and λ_k in (3.10) is fixed to 1. The initial labels for G-Net remain unchanged.
- G/S/D-Net+ACL uses adjustable counting loss (ACL) when training G/S/D-Net, that is, λ_k in (3.10) increases for every epoch as described in Section 3.3.4.

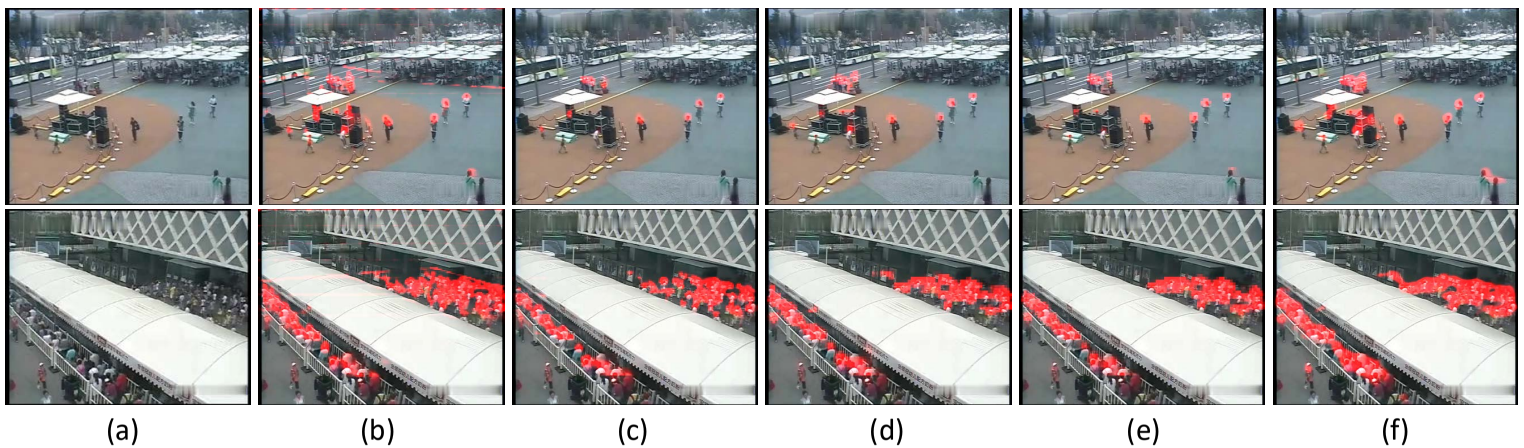


Figure 3.10: **Example density blending images of self-evaluation** (a) Input image, (b) Baseline, (c) G/S/D-Net, (d) G/S/D-Net+ACL, (e) G/S/D-Net+ACL+RP, and (f) SEN-ACL (Proposed)

- G/S/D-Net+ACL+RP applies a re-labeling process (RP) described in 1 when training G/S/D-Net+ACL.

- SEN-ACL (**Proposed**) includes R-Net in G/S/D-Net+ACL+RP.

Counting accuracy comparison for the various configurations above is compared in Table 5.3. The self-evaluation findings were as follows: 1) The baseline network is similar to the structure of [4] but differs in terms of the additional counting loss. When counting loss is simply added to density loss at a fixed rate, baseline counting accuracy is lower than that of [4] which archived 12.9 MAE on that dataset. 2) The networks that learned differently by dividing crowdness showed reduce the counting error as compared to the baseline, but the performance was still lower than [4]. This is because, unlike [4], G/S/D-Net is not fine-tuned according to the target scene. 3) When the weight of the counting loss was gradually increased rather than remaining fixed, counting performance improved to state-of-the-art level. This confirms that it is more effective to adjust the weight of counting loss according to crowdness and epochs than to simply add the counting loss. 4) As the initial *sparse* or *dense* label used in training G-Net was inaccurate, additional enhancement was achieved by re-labeling process. 5) R-Net effectively reduced the errors that occur when resizing images and aggregating patches for the initial density map.

Figure 3.10 presents examples of qualitative estimation results achieved by various self-evaluation settings. The baseline network has griding artifacts and discontinuity in density values as depicted in Figure 3.10-(b), which is caused by aggregating local patch crowd density maps. Separating the degree of crowdness in the network structure can mitigate griding artifacts as shown in Figure 3.10-(c). In Figure 3.10-(d,e), additional counting loss and re-labeling process can improve the counting accuracy though there is blurry effect in the crowded region. Finally, The crowd density map is refined to higher resolution compared to the other settings with a help of R-Net as depicted in Figure 3.10-(f).

Table 3.4: Comparison errors on WorldExpo'10 [4]

Method	S1	S2	S3	S4	S5	Average MAE
CCNN [4]	9.8	14.1	14.3	22.2	3.7	12.9
MCNN [5]	3.4	20.6	12.9	13.0	8.1	11.6
Switch-CNN [42]	4.2	15.7	**10.0**	11.0	5.9	9.4
ConvLSTM [20]	6.8	14.5	14.9	13.5	3.1	10.6
SEN-ACL (Init. density map)	4.9	15.9	13.8	**10.2**	6.9	10.3
SEN-ACL	**2.7**	**13.0**	10.5	16.1	**2.9**	**9.0**

Table 3.5: Comparison errors on UCSD [13]

Method	MAE	MSE
KRR [1]	2.16	7.45
CA-RR [14]	2.07	6.86
CCNN [4]	1.60	3.31
MCNN [5]	**1.07**	**1.35**
Hydra-CNN [2]	1.51	-
Switch-CNN [42]	1.62	2.10
ConvLSTM [20]	1.13	1.43
SEN-ACL	1.39	1.72

3.4.4 Comparative Evaluation with State of the Art Methods

The WorldExpo'10 dataset was first introduced in [4]. We conducted the experiment with Zhang *et al.*'s settings [4]. The results in Table 3.4 show that the proposed method achieves the best performance in terms of average MAE in Scenes 1, 2 and 5, and competitive performance in the remaining scenes, as compared with recent deep learning-based methods (CCNN [4], MCNN [5], Switch-CNN [42] and ConvLSTM [20]).

For the UCSD dataset, we followed the experimental settings in [13]. As the UCSD dataset is a low-resolution video, it does not have enough room for 3 pooling layers for G/S/D-Net, so we excluded one pooling layer and *conv 4* layer in the proposed

Table 3.6: Comparison errors on Mall [1]

Method	MAE	MSE
KRR [1]	3.15	3.96
CA-RR [14]	3.43	4.21
COUNT Forest [16]	2.50	-
ConvLSTM [20]	<u>2.10</u>	**2.76**
SEN-ACL	**2.09**	<u>2.78</u>

network structure. The results of the evaluation are summarized in Table 3.5. The compared methods are hand-crafted feature regression-based methods (Kernel Ridge Regression (KRR) [1], Cumulative Attribute Regression (CA-RR) [14]), and deep learning-based methods (CCNN [4], MCNN [5], Hydra-CNN [2], Switch-CNN [42] and ConvLSTM [20]). The proposed method was shown to perform competitively against state-of-the-art methods. In the case of UCSD dataset, there are little difference between sparse and dense density regions so that we got a little improvement over the single network like CCNN [4]. Also, since UCSD dataset was collected in real-time surveillance camera, it has high temporal consistency, which is suited to recurrent networks like ConvLSTM [20].

In the Mall dataset, we used the same experimental settings [1]. The results are reported in Table 3.6. The compared methods are Kernel Ridge Regression (KRR) [1], Cumulative Attribute Regression (CA-RR) [14], and COUNT Forest [16], which are hand-crafted feature regression-based methods; and ConvLSTM [20], which is a deep learning-based method. The proposed method achieved best performance on MAE metric and second best performance on MSE metric.

Note that, in the UCSD and Mall experiments, the value of λ (representing a balance of density map accuracy and counting accuracy) was of the same as in WorldExpo'10. Because these datasets have different scene characteristics (such as number of people and degree of crowdness), more improvement can be achieved by adjusting λ for each

(a) Scene 1 (b) Scene 2 (c) Scene 3 (d) Scene 4 (e) Scene 5

Figure 3.11: **Example frame images of ShanghaiTech Part A dataset.** If the dominant crowd sparsity is denoted by *S* (*Sparse*) or *D* (*Dense*), ShanghaiTech Part A dataset can be consisted of (a) Scene 1 (*S*), (b) Scene 2 (*D*), (c) Scene 3 (*S, D*), (d) Scene 4 (*D*), and (e) Scene 5 (*S*)

dataset.

3.4.5 Analysis on the Proposed Components

We additionally conducted an experiment to verify each elements of the proposed network as a motivation in this chapter. The experiment was conducted on ShanghaiTech Part A dataset. In subsets of ShanghaiTech Part A dataset, Scenes 1-5, the dominant sparsities of crowd are as shown in Fig. 3.11. As summerized in Table 3.7, the counting loss is effective for a dense crowd as dense crowded scenes (*e.g.* Scene 2 and Scene 3) were improved than the prior version, and the separation of S, D-Net is effective for a sparse crowd as sparse crowded scenes (*e.g.* Scene 1 and Scene 5) were improved than the prior version. In addition, as shown the results between version 4 and 5, the proposed re-labelling process and R-Net improved the performance in all cases regardless of the degree of congestion of crowd.

3.5 Summary

In this chapter, we proposes a novel CNN architecture for crowd density estimation that selectively utilizes sub-networks with respect to crowdness. We also propose an adjustable loss scheme for each sub-network that adjusts the balance of counting loss

Table 3.7: Experimental results of in-detail analysis of the proposed components on the ShanhaiTech Part A dataset. **bolded** numbers are denoted effectively improved results

Ver.	S1	S2	S3	S4	S5	Mean	Description
1	6.80	51.30	27.50	13.60	8.60	21.54	Baseline
2	14.20	**32.60**	**9.90**	16.90	11.10	17.00 (-4.54)	Added count loss (effective for dense crowd patches)
3	**5.10**	21.40	15.10	19.30	**4.40**	13.10 (-3.90)	Added S, D-Net (effective for sparse crowd patches)
4	4.90	**15.90**	13.80	**10.20**	6.90	10.30 (-2.80)	Added re-labelling process
5	**2.70**	**13.00**	**10.50**	16.10	**2.90**	9.00 (-1.30)	Added R-Net

and density loss, depending on crowdness and training epochs. This adjustable loss scheme can also handle the scale issue in which high-density regions are predominantly learned. In addition, the proposed refinement sub-network effectively renders the density map as high resolution map by taking account of contextual information. To the best of our knowledge, this is the first attempt to resolve the trade-off between density map accuracy and counting accuracy by considering both network architecture and loss functions. As the comparative evaluation shows, our network exhibits state-of-the-art performance for three publicly available datasets. The self-evaluation results confirm the validity of the components of the proposed method (selective ensemble, adjustable loss, alternating re-labeling, and refinement sub-network).

Chapter 4

Sequential Crowd Density Estimation from Center to Periphery of Crowd

4.1 Overview

The goal of crowd density estimation (CDE) task is to get the number of people or to get their distribution from images acquired from a wide-view camera such as the surveillance camera. Recently, the introduction of the intelligent surveillance system and the growing interests in preventing terrorism leads to active research on the CDE. However, the CDE is one of the most challenging computer vision tasks, because 1) the occlusion is so severe that conventional detection methods (*e.g.* head or pedestrian detection) cannot accurately count crowd, and 2) it is difficult to represent the features of the crowd, which varies according to the various appearance of people and the degree of congestion of the scene. Recently, with the impressive development of the deep learning, many methods based on the convolutional neural network (CNN) have been proposed with the state-of-the-art performance.

The CNN-based CDE methods [2, 4, 5, 18–22, 62] conventionally estimate a crowd density map where a person is represented as a small Gaussian kernel with a sum of 1. That is, the summation of the crowd density map becomes the number of people in the

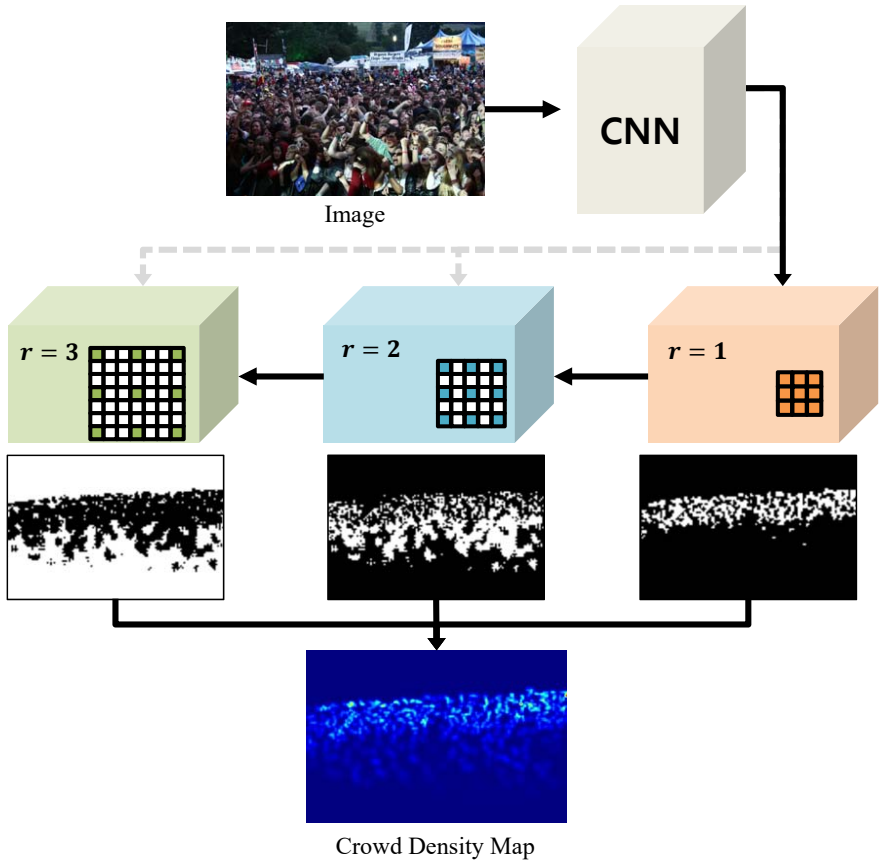


Figure 4.1: **An illustration of the proposed method** In this work, we have confirmed that accurate crowd density can be obtained by estimating the center and periphery of each person separately. The proposed network gradually estimates crowd density from center to periphery of the crowd by each dilated CNN and integrates the results.

entire image. To estimate the accurate crowd density map, we gradually estimate the Gaussian kernel from its center to the periphery. The center of the Gaussian kernel can be obtained by only considering a small region, while the other regions are easy to be combined with neighbors so that it can be obtained with a wide field of view. From the viewpoint of the network, the field of view of a network is defined by receptive field, which implies that the network can better estimate the density map of a person if it has various receptive fields. However, it is not desirable to train multiple neural networks separately depending on the receptive fields.

To realize the integration of the various receptive field, we propose a novel CNN structure using multiple dilated CNN blocks that have a variety of receptive fields and is trained in cascading from small-scale dilated CNN block to large-scale dilated one. As shown in Figure 4.1, the feature extracting CNN is shared and its following multiple CNN blocks contain a dilated convolutional layer with different filter sizes. By utilizing the dilated convolutional layers with different filter sizes, the receptive fields of the multiple dilated-CNN blocks can avoid the overlapping. In addition, the memory usage can be much reduced in spite of the multiple blocks. The multiple blocks are working in cascading to estimate the crowd density map gradually from center to periphery of each person area, and the final crowd density map is obtained by mutually integrating all the results from the multiple network branches. We validate our proposed method in real-world benchmark datasets, which shows the state-of-the-art performance for the various surveillance environments.

In summary, the contribution of the proposed work is as follows,

- We propose a novel crowd density estimation method, which estimates each Gaussian kernel for each person from center to periphery progressively.
- To mutually learn the different regions for the Gaussian kernels, we propose a novel CNN structure using multiple dilated-CNN blocks with different scales. The final crowd density map is obtained by integrating the results of the multiple dilated-CNN blocks.

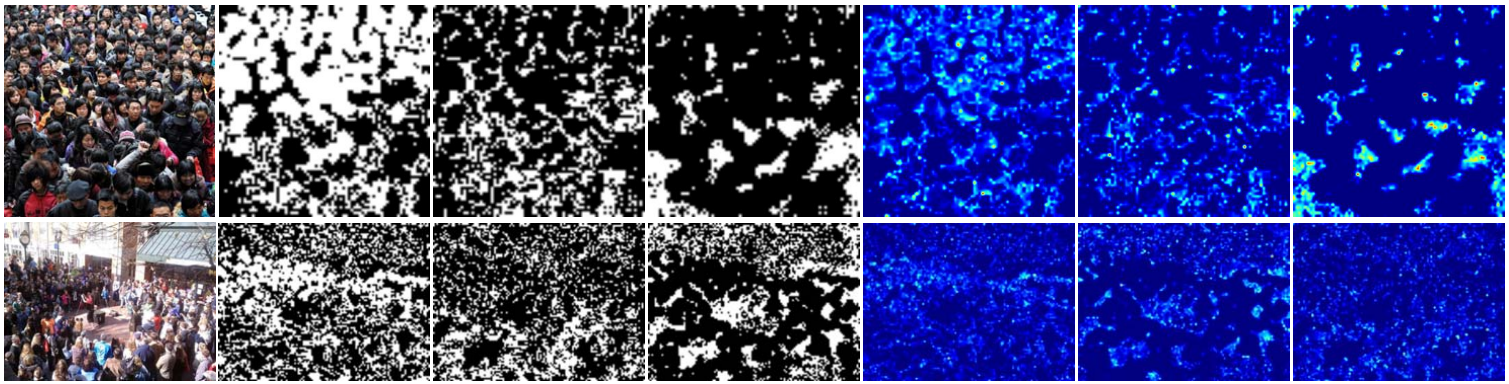


Figure 4.2: **Simple analysis on relation between estimation accuracy using the network and its dilated rates** When (a) the input image is given, (b-d) indicate the region with the lowest error (i.e., the region where the density estimated accurately) among density maps estimated by the network with the dilated rate 1, 2, and 8. (e-g) show the intensity of the estimation error. We confirm that the density map in the periphery region of the crowd can be accurately estimated using the network with a large receptive field.

- In the experiments conducting on the challenging datasets including Shanghaitech-PartA,B and UCF_CC_50, our proposed method achieves the state-of-the-art performance.

4.2 Cascade Residual Dilated Network (CRDN)

In this section, we describe our proposed method named Cascade Residual Dilated Network (CRDN). We first demonstrate the relationship between dilated convolution and its effect on the receptive field. Then, we analyze how the multiple receptive fields effect on estimating crowd density. Along with the above analysis, we describe how our proposed method estimates accurate crowd density in detail.

4.2.1 Effects of Dilated Convolution in Crowd Counting

The 2D-dilated convolution is defined as following,

$$y(m,n) = \sum_{i=1}^{M} \sum_{j=1}^{N} x(m + r \cdot i, n + r \cdot j) \cdot w(i,j), \tag{4.1}$$

where $y(m,n)$ is the output obtained from the input $x(\cdot,\cdot)$ and the weight $w(\cdot,\cdot)$ with the dilated rate r. The basic 2D convolution is the case of $r = 1$, that is, dilated convolution functions to enlarge the convolution operation with offset r. Dilated convolution is one of the key factors that improve the accuracy of semantic segmentation recently [63–68]. With the sparsely arranged kernel values, dilated convolution can enlarge the receptive field of the neural network without increments of computational complexity, and also can replace conventional pooling layers. As a result, the dilated convolutional network is able to represent contextual information without the loss of the resolution.

Figure 4.2-(b-d) is a binary map of the region where the correct prediction is made according to the dilate rate. Note that with a wide range of contextual information (e.g. with the network of large receptive field), the estimation of crowd density in the

inter-personal region (also region that farther from neighboring people) is more accurate than its counterpart. In a wide field of view, it can be seen that the density map in a region between densely crowded regions is better estimated by the network with a large receptive field. We have confirmed that by changing the receptive field using dilated convolution, the amount of contextual information can be effectively controlled. Figure 4.2-(e-g) show how accurately the crowd density is predicted in each region. Each region is represented by being normalized after a division by the maximum value of the error among the crowd density maps estimated by the network with dilated rates 1, 2, and 8. As shown in the figure, when the dilated rate increases, the density map of the region corresponding to the periphery of the crowd is accurately estimated. The crowd density map is generated by combining the density map of each individual. In general, the density map of each person is expressed by a Gaussian kernel with a standard deviation depending on the scale of a person. Therefore, the density value is inversely proportional to the distance from the center of the person. We can determine the center of individuals with a distribution of the crowd density, and progressively the periphery region is also designated.

4.2.2 The Proposed Network

Founded on the above finding and empirical cues, we propose a novel way to estimating crowd density from center to the periphery of the crowd. As depicted in Figure 4.3, our proposed network consists of Multi-Dilated Convolutional Blocks (MDBs) with a various dilated rate. Each MDB estimates the crowd density under the guidance of adjacent MDBs. In particular, each MDB is connected in cascades to its next block so that all MDBs can be learned complementarily. In detail, the proposed network is composed of two parts; Frontend and MDBs. The Frontend network is a fully convolutional network acted as a feature extractor, which extracts high-level feature maps from the RGB input image. The feature map extracted from the Frontend is then fed to MDBs. MDB consists of convolution layers with pre-defined dilated rate

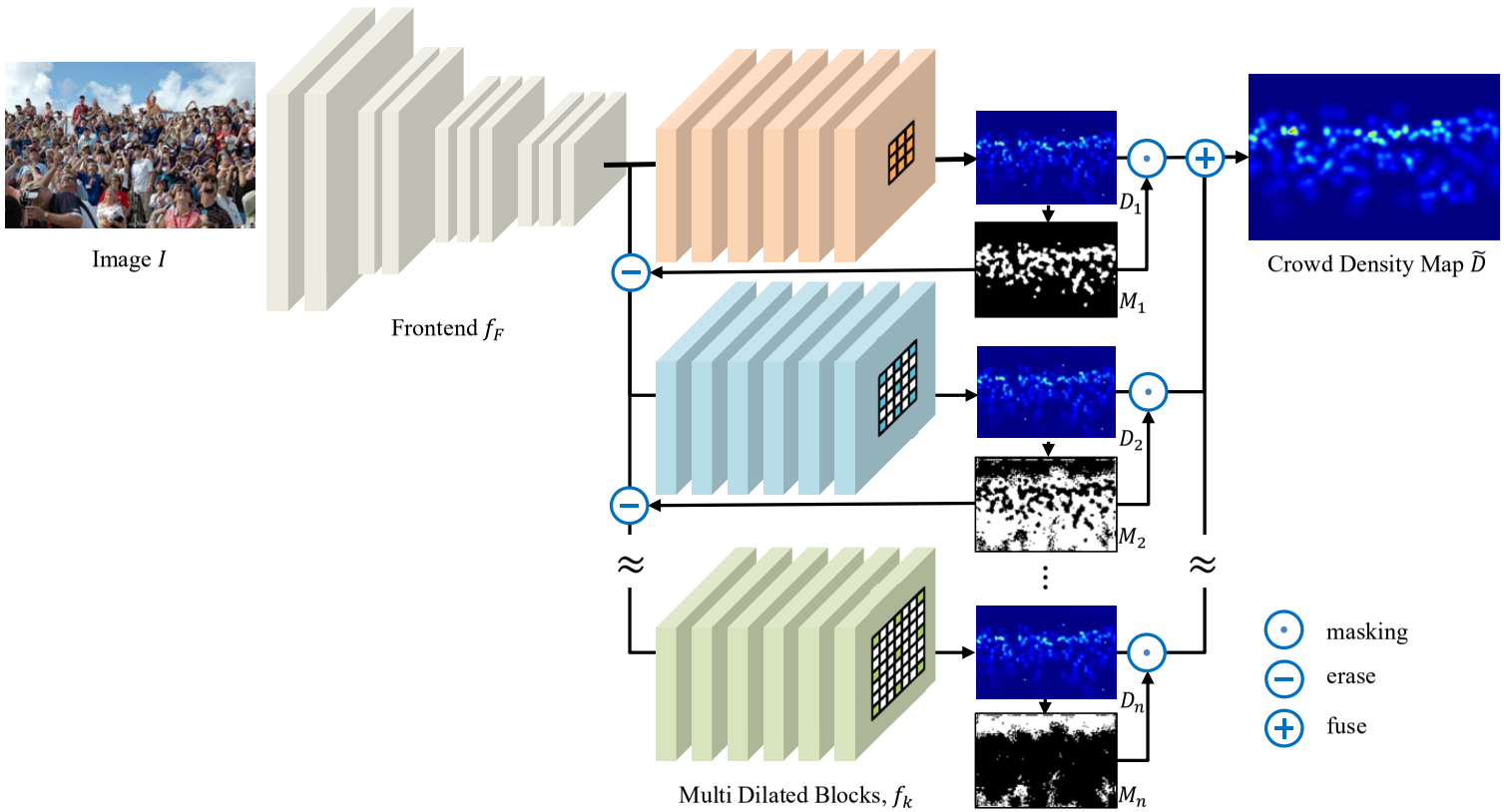


Figure 4.3: **Overall framework of the proposed network** The input feature maps are extracted by the frontend network f_F. The extracted features are fed as input to each Multi-Dilated Blocks (MDBs) ($f_1 \cdots f_n$) to estimate the crowd density corresponding to each MDB. In each MDB, only the density in an attention region becomes the final crowd density. From the second stage, the crowd density is estimated using the feature map excluding attention regions of the previous MDBs.

and is followed by 1×1 convolution to form the crowd density map. Crowd density is progressively estimated by MDBs from the center to periphery of the crowd. Formally, partial crowd density map $D_k(k = 1, ..., n-1)$ is estimated by each MDB B_k with the dilated rate r_k. At each step, only a partial attention region M_k of D_k is left by masking as follows,

$$\tilde{D}_k(m, n) = D_k(m, n) \odot M_k(m, n), \tag{4.2}$$

where M_k is binary mask indicating the region left in k-th stage and $\tilde{D}_k$ is final crowd density map of M_k. In the first stage, M_1 indicates the most crowded region which has density values of greater than δ_1 after the normalization. The other partial regions M_k except for the n-th stage indicate regions with intermediate density values of less than δ_{k-1} and greater than δ_k. The remaining region M_n which has the least crowded is the region with density values of less than δ_{n-1}. Each partial attention region also effects on the feature maps, which in next step erases the feature maps corresponding to the partial attention map for complementary learning. By combining estimated regions at all stages, we get the final crowd density map of the whole image. The above procedure is summarized in Algorithm 2, where $\bar{D}$ indicates normalized density map.

Thresholds as the guidance to the input feature map in each MDBs are determined according to the characteristics of the scene. In general, a crowd density map is defined by convolving Gaussian kernel on dot annotation map of the people's head locations so that the range of density values is varied depending on the level of crowded of the scene. We conduct normalization process and hard thresholding to represent attention region M_k regardless of the crowed level of the scene. Additionally, because the standard deviation σ in Gaussian kernel for generating density maps is given depending on the scene, we need a common rule to generalize the attention regions of each MDB. We utilize reinforcement learning as post-processing to set thresholds for each dataset. In this work, because of its low-dimentionality of thresholds, we use REINFORCE algorithm [69] which simply uses immediate rewards to estimate the value of the policy to determine the sub-optimal thresholds. We define a state as current threshold values,

Algorithm 2: Estimation process of the proposed network

Data: Image I and thresholds $\delta_1, ..., \delta_{n-1}$

Result: Crowd density map of I, $\tilde{D}$

$\Phi \leftarrow f_F(I; \theta_F)$

$D_1 \leftarrow f_1(\Phi; \theta_1)$

$M_1 \leftarrow I[\bar{D}_1 > \delta_1]$

$\tilde{D}_1 \leftarrow D_1 \odot M_1$

$\tilde{\Phi} \leftarrow \Phi \odot (\mathbb{I} - M_1)$

for $k = 2$ ***to*** $n - 1$ **do**

  $D_k \leftarrow f_k(\tilde{\Phi}; \theta_k)$

  $M_k \leftarrow I[\delta_{k-1} > \bar{D}_k > \delta_k]$

  $\tilde{D}_k \leftarrow D_k \odot M_k$

  $\tilde{\Phi} \leftarrow \tilde{\Phi} \odot \bigcap_{j=1}^{k} (\mathbb{I} - M_j)$

end

$D_n \leftarrow f_n(\tilde{\Phi}; \theta_n)$

$M_n \leftarrow I[\delta_{n-1} > \bar{D}_n]$

$\tilde{D}_n \leftarrow D_n \odot M_n$

$\tilde{D} \leftarrow \sum_{i=1}^{n} \tilde{D}_i$

an action as the difference of thresholds from its initial values, and a reward as a change of the Mean Absolute Error (MAE) on validation samples. The actor is implemented by multilayer perceptron with hidden nodes h_{RL}, which is parameterized θ_{RL}. We train the actor network by stochastic gradient ascent [69] to maximize the expected gain of MAE as follows,

$$\Delta\theta_{actor} \propto \sum_{t=1}^{T} \frac{\partial log\ p(a_t|s_t; \theta_{actor})}{\partial \theta_{actor}} r_t, \tag{4.3}$$

where $p(a_t|s_t)$ denotes the conditional action probability, T is the maximum number of trial, and r_t is the reward at each trial.

4.3 Experiments

4.3.1 Datasets and Experimental Settings

We have evaluated the proposed method on three challenging crowd counting datasets:

• **ShanghaiTech dataset Part A, B [5]** contains 1,198 images with a total of 330,165 people and is divided into two parts: Part A containing 482 images of congested scenes (300 images for training and 182 images for testing) and Part B containing 716 images of sparse scene (400 images for training and 316 for testing).

• **UCF_CC_50 dataset [70]** contains 50 images downloaded from web. The number of people per image ranges from 94 to 4,543 with an average of 1,280 individuals. Its small number of images and large variance of people numbers make this dataset very challenging. We use 5-fold-cross-validation setting as described in [70].

To evaluate our proposed method, we use both Mean Absolute Error (MAE) and Mean Squared Error (MSE) as evaluation metrics.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|c_i - c_i'\right|, MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|c_i - c_i'|^2}, \tag{4.4}$$

where N is the number of test images, c_i is the number of people in the i-th image, and c_i' is the estimated number of c_i. The number of people in the image is obtained by integration of the crowd density over whole image regions as follows,

$$c_i = \sum_{m=1}^{H}\sum_{n=1}^{W} D^{GT}(m,n). \tag{4.5}$$

The estimated case is similar way of Eq.(5.13).

4.3.2 Implementation Details

As a conventional way, we generate the ground truth of crowd density map by blurring dot annotation map on head locations using a Gaussian kernel with fixed standard deviation.

$$D_I^{GT}(x_i) = \sum_{i=1}^{c_I}\delta(x - x_i) * G_\sigma(x), \tag{4.6}$$

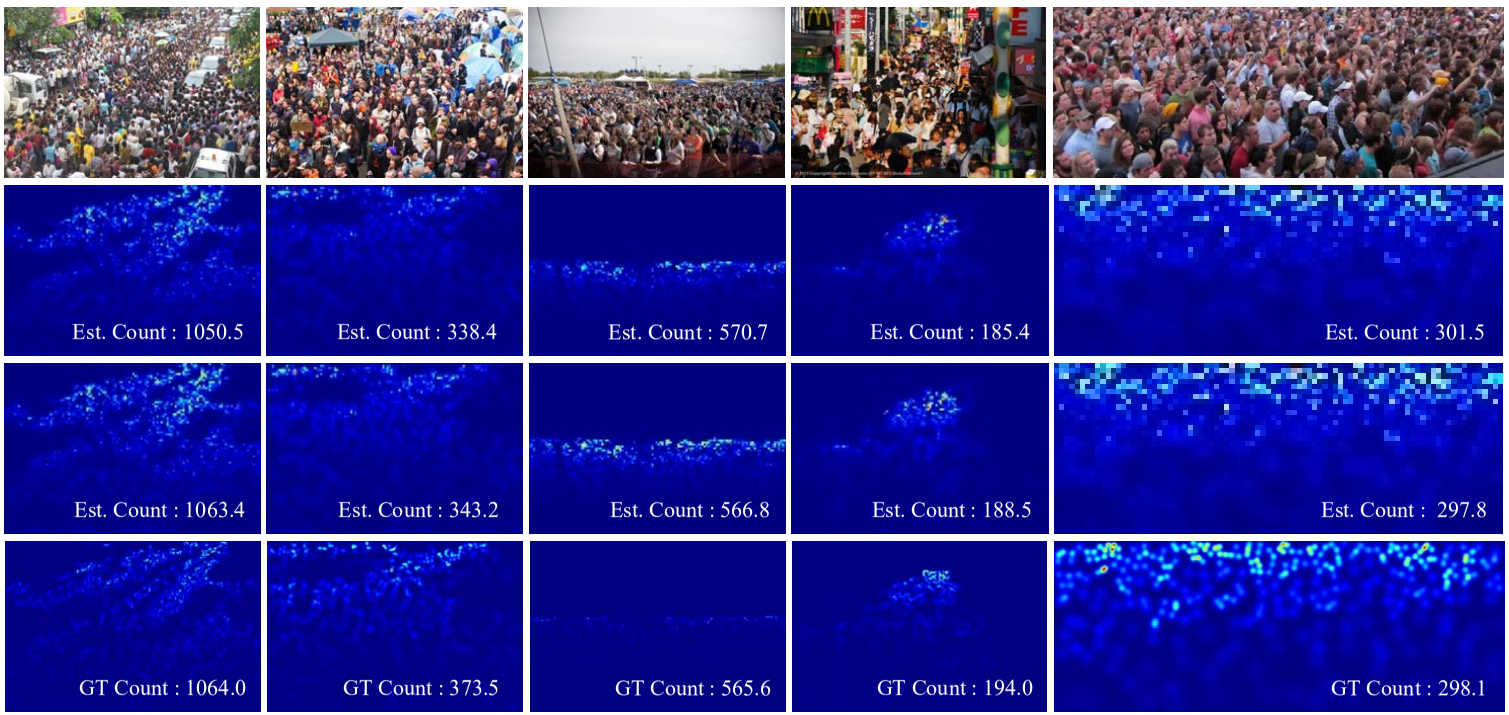


Figure 4.4: **Qualitative results on ShanghaiTech-Part A dataset** The first row is an input image, the second row is the estimated density map from CSRNet [21], the third row is estimated density map from the proposed method, and the last row is the ground truth.

Table 4.1: Network structure of the proposed network (the configuration of convolution layer is briefly expressed as *[kernel size]-[number of channels]-[dilated rate]*.)

Layer	Frontend	Layer	Backend (MDBs)
1-1,2	conv3-64-1	1-1	conv3-512-r
	max pool	1-2	conv3-512-r
2-1,2	conv3-128-1	1-3	conv3-512-r
	max pool	2	conv3-256-r
3-1,2,3	conv3-256-1	3	conv3-128-r
	max pool	4	conv3-64-r
4-1,2,3	conv3-512-1	5	conv1-1-1

where x_i is annotation point of head location, G_σ is a Gaussian kernel whose standard deviation is σ, and $*$ indicates the convolution operation. We set σ as 15 in ShanghaiTech-Part B. In some dataset with high congested scenes such as ShanghaiTech-Part A and UCF_CC_50, we use a geometry-adaptive kernel [5] which is a kind of a Gaussian kernel with variable standard deviation depending on the person's location, $\sigma_i = \beta \bar{d}_i$, where $\bar{d}_i$ is the average distance of k nearest neighbors of i-th person. We use $\beta = 0.3$ and $k = 3$ as in [5].

We use the front part of VGG16 [71] as the Frontend of the proposed CRDN. Specifically, the first 10 layers of VGG16 with three pooling layers are configured as the Frontend. As the output density map is shrunk due to pooling layers, we expand the resulted density map by bilinear interpolation of factor 8. As described in the Sec. 4.2.2, the input of each MDB except for B_1 is the erased feature map which is extracted from the Frontend. The erasing mask is a binary mask that normalizes the density map results from the previous MDB to a range of 0 to 1 and performs hard thresholding. The network configuration of the Frontend and MDBs is summarized in Table 5.1.

We use dilated rates $r_k = \{1, 1.5, 2, 4, 8\}$ in the experiment. $r = 1.5$ means that dilated rates of first three layers are set to 1 and those of the others are set to 2. The loss

Table 4.2: Estimation errors on ShanghaiTech and UCF_CC_50 dataset

	SHT Part A		SHT Part B		UCF_CC_50	
Method	MAE	MSE	MAE	MSE	MAE	MSE
CCNN [4]	181.8	277.7	32.8	49.8	467.0	498.5
MCNN [5]	110.2	173.2	26.4	41.3	377.6	509.1
SCNN [19]	90.4	135.0	21.6	33.4	318.1	439.2
CPCNN [23]	73.6	106.4	20.1	30.1	298.8	320.9
ACSCP [22]	75.7	102.7	17.2	27.4	291.0	404.6
CSRNet [21]	68.2	115.0	10.6	16.0	266.1	397.5
ic-CNN [27]	68.5	116.2	10.7	16.0	260.9	365.5
SANet [28]	67.0	104.5	**8.4**	**13.6**	258.4	334.9
Ours	**65.2**	**103.7**	10.7	18.2	**230.9**	**326.3**

used for training CRDN is L_2 loss as follows:

$$L_2 = \frac{1}{N}\sum_{i=1}^{N}\left\|D_i^{GT} - \tilde{D}_i\right\|_2 \tag{4.7}$$

The Adam optimizer is used, and the learning rate is attenuated 0.1 times at $1/3$ and $2/3$ over $1e^{-6}$. For the quick convergence of the training process, we first train each branch separately, and then we fine-tuned with the whole structure of CRDN. When training each branch, the input of the Backend is the overall feature map and the loss Eq.(4.7) of the output density map is used.

4.3.3 Comparison with Other Methods

Shanghaitech Dataset ShanghaiTech-Part A consists of images randomly crawled from the Internet. Since there are many challenging samples such as synthetic, gray-scale, or watermarked images, it is suitable to compare the robustness of counting accuracy. Different from Part A, images from Part B is taken from the streets in Shanghai so

that these have similar scene characteristics such as camera installation environments, and there is a relatively sparse crowd. Experimental results with eight recently work is shown in Table 4.2. In both parts of the dataset, our proposed algorithm achieves the lowest error in MAE compared to other methods and we get lower MAE than the state of the art SANet [28]. Compared to CSRNet [21] which consists of a single dilated convolutional network, it is confirmed that our proposed multi-dilated network and its configuration improves the performance. Qualitative results are depicted in Figure 4.4. The results show that the distribution of density values are similar to those of CSRNet with some density values are enhanced to accurate, which show that each part of the Gaussian kernel is accurately estimated.

UCF_CC_50 Dataset UCF_CC_50 dataset is a relatively small dataset consisting of only 50 gray-scale images. Despite this, images of UCF_CC_50 dataset represent hundreds to thousands of people, which can cause over-fitting of the network. To prevent over-fitting due to the small size of the dataset, we fine-tuned the network from the pre-trained one on ShanghaiTech-Part A. The counting performance comparison with the eight recently reported algorithms is summarized in Table 4.2. The proposed method showed the lowest MAE of 230.9 and also showed the lowest MSE of 326.3. The qualitative results are shown in Figure 5.6.

4.3.4 Ablation Study

Comparison between Multi-Dilated Convolutional Blocks

We conducted an experiment to see how each of the proposed MDBs contributed to overall performance. Table 4.3 summarizes the results of comparing the performance of each MDB and the average of the density maps estimated by all MDBs. The results show that when a single network is configured, the best performance is achieved when the dilated rate is 2. This result is the same as reported by in [21]. One interesting thing is that the images which achieve the best counting performances among MDBs

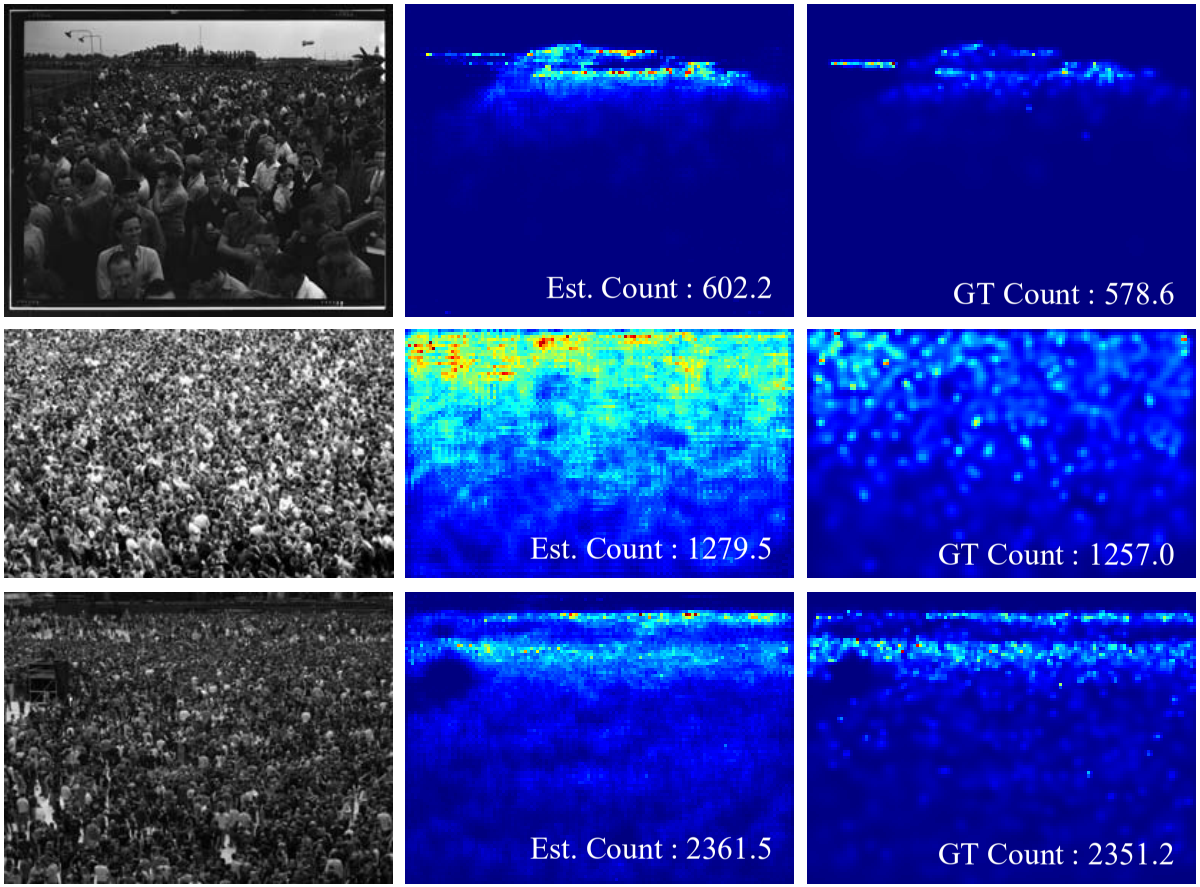


Figure 4.5: **Qualitative results on UCF_CC_50 dataset** The first column is the input image, the second column is the estimation result, and the third column is the ground truth of the crowd density map. The density map is normalized for clear visualization.

Table 4.3: Estimation errors from ablation study (*Oracle* chooses the best result from each MDB.)

MDB-1	MDB-1.5	MDB-2	MDB-4	MDB-8	AVG	*Oracle**	MAE	MSE
✓							67.2	**103.7**
	✓						70.5	107.8
		✓					**65.9**	104.8
			✓				71.1	106.8
				✓			293.6	402.9
✓	✓	✓	✓		✓		78.2	127.8
✓	✓	✓	✓			✓	**46.8**	**78.4**
✓	✓						70.7	108.2
✓	✓	✓					65.8	**103.6**
✓	✓	✓	✓				**65.2**	103.7
✓	✓	✓	✓	✓			**65.2**	103.7

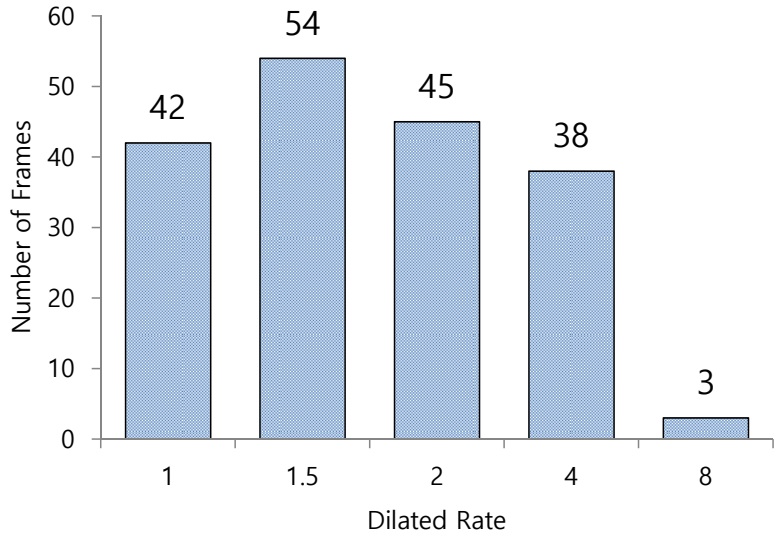


Figure 4.6: Distribution of the images which achieved the best performances among MDBs on ShanghaiTech-Part A dataset

are almost uniformly distributed except for the case of dilated rate 8, as shown in Figure 4.6. Also, in the view of network selection, when we select the MDB which is best performed with the image, MAE is reduced to 46.8, which is denoted by *Oracle* in the table. We tested the case selecting a network with a classification network, but we did not get a meaningful result. The reason is that there is no distinct correlation between the contents of an image and the receptive field improving counting accuracy.

Configuration of Combination of Multi-Dilated Convolutional Blocks

We measured the performance with the addition of an MDB with a specific dilated rate. The results summarized at the bottom of Table 4.3 show that the best performance is achieved when $\{1, 1.5, 2, 4\}$ MDBs are considered, and there is no performance improvement when MDB with a dilated rate greater than 4 is added. This is because there is a trade-off between the receptive field of the network and the accuracy of the density estimation, which show that as the receptive field of the network larger, the final density estimation results in the lower performance because of using the sparse information of the image.

Determining Residual Thresholds

In order to tune the residual thresholds δ in Section-4.2.2, we use reinforcement learning (RL) as post-processing. The action space is defined in a continuous real domain where the action means the update value of the threshold with a fixed gap (± 0.001 in this chapter). The reward is set to the inverse of value changes of MAE on the validation set. As depicted in Fig. 4.7, the trained agent adjusts the threshold to decrease MAE and the threshold is set to the sub-optimal values in eight time steps. Applying the RL as a post-processing improves the performance from 66.8 to 65.2 on MAE in the Shanghaitech Part A dataset.

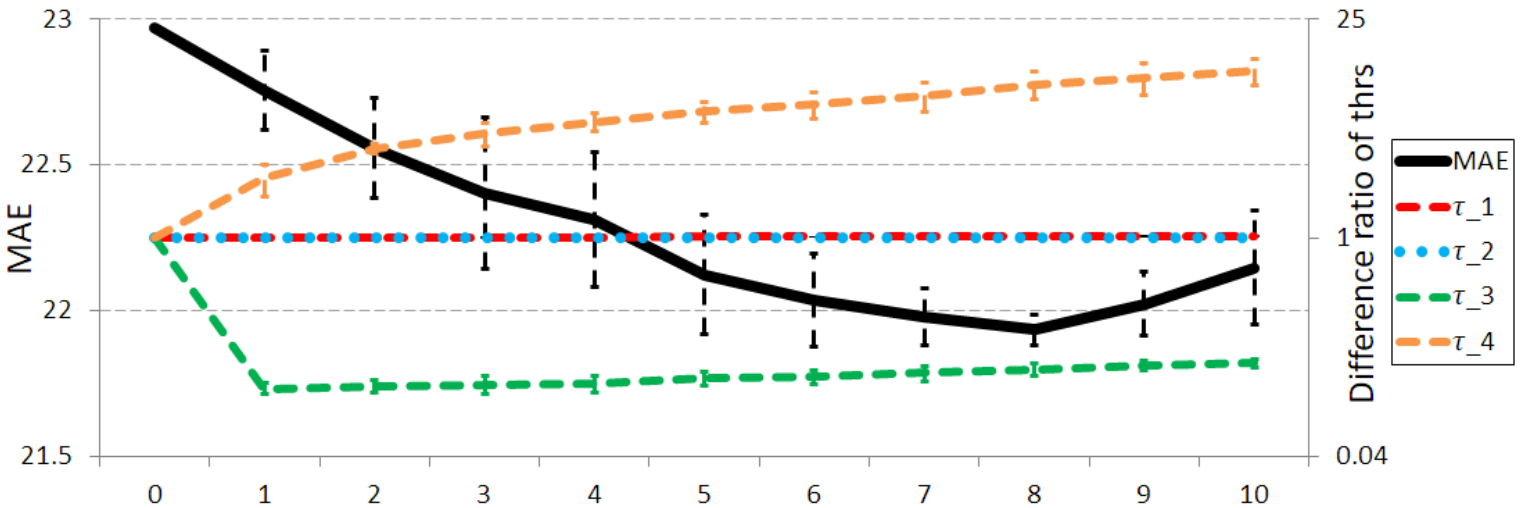


Figure 4.7: Experiment on setting residual thresholds

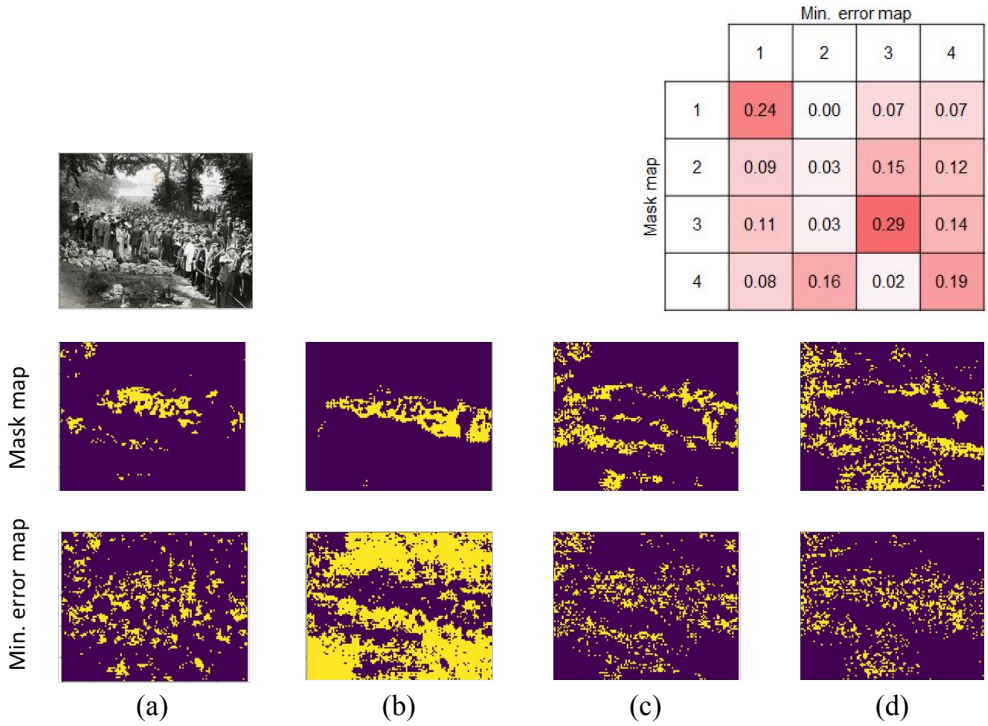


Figure 4.8: **Example figures for in-detail analysis (1).** Given an input image, (a-d) show both mask map M_i and minimum error map of each MDB. The minimum error map indicates region that lowest counting error among density estimation results of MDBs. The table shows the correlation scores calculated by Intersection over Union (IOU) of both mask map M and minimum error map of each MDB.

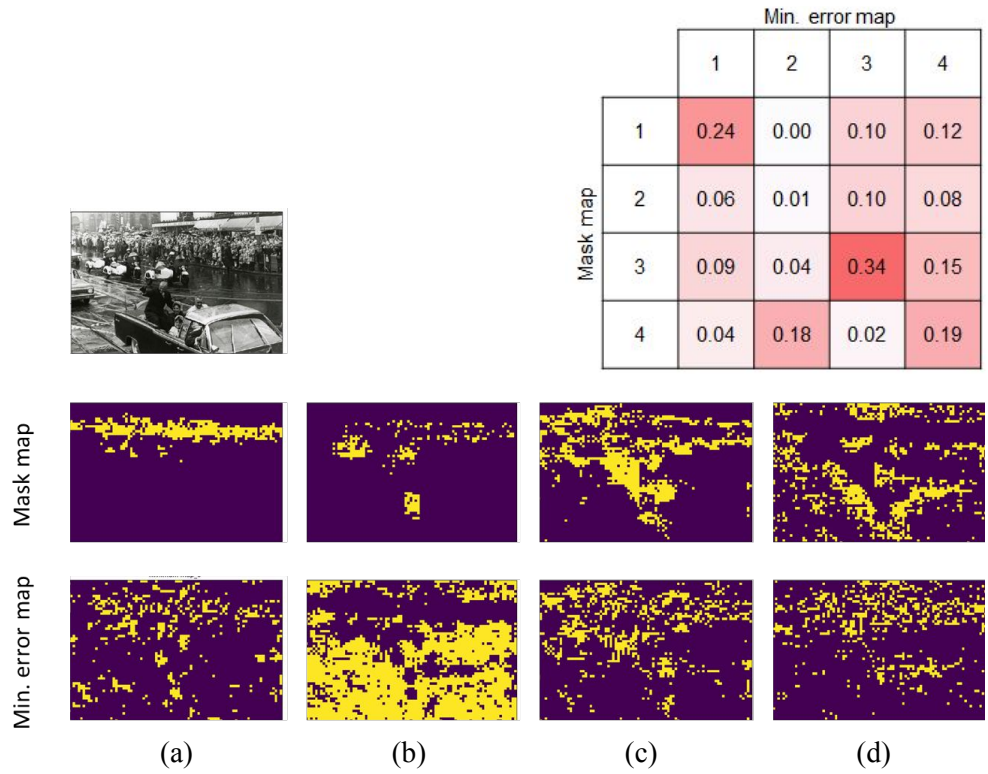


Figure 4.9: **Example figures for in-detail analysis (2).** Given an input image, (a-d) show both mask map M_i and minimum error map of each MDB. The minimum error map indicates region that lowest counting error among density estimation results of MDBs. The table shows the correlation scores calculated by Intersection over Union (IOU) of both mask map M and minimum error map of each MDB.

4.3.5 Analysis on the Proposed Components

Additional experiments were conducted to confirm the effectiveness of the proposed element in this chapter. Experiments were conducted on the ShanhaiTech Part A dataset. As depicted in Figs. 4.8, 4.9, we illustrated the mask map and minimum error map of each MDB, and then calculated correlation values between the two maps. The minimum error map indicate the region with the lowest counting error among the density estimation results of MDBs. The correlation values between the minimum error map and the mask maps was calculated by the Intersection-over-Union (IoU) score and summarized in the tables of the Figs. 4.8, 4.9. Through the experiments, it was confirmed that the correlation between the mask map and the region estimated accurately by each MDB was high except for the case of MDB-1.5 The reason is that, as can be seen from Fig. 4.6, MDB-1.5 has the highest performance in the large number of sample images.

4.4 Conclusion

In this chapter, we proposed a novel crowd density estimation method which gradually estimates density from the center to periphery of each person. We first showed an empirical finding that the accuracy of the density estimation for the centered or surrounding region of individuals depends on the scale of dilated convolution. The centered region can be estimated well by a small-scaled dilated convolution, while a large-scaled dilated convolution makes small error for the surroundings. Based on the finding, we proposed Cascade Reidual Dilated Network (CRDN) equipped with multiple dilated CNN blocks. CRDN estimates the density of the center of each person with the small-scale dilated CNN block first, and then subsequently estimates the remaining areas with the larger-scale blocks. Extensive experiments show that the proposed method can accurately estimate crowd density over the state-of-the-art.

Chapter 5

Congestion-aware Bayesian Loss for Crowd Counting

5.1 Overview

Crowd density estimation can be accomplished with a computer vision-based algorithm to count the number of people in an image, which is one of the challenging tasks for an intelligent surveillance system. Using a crowd density estimation algorithm, we can determine regions of interest where crowds are forming. We can then reduce the computational resources of various algorithms of surveillance system [6, 72, 73] by concentrating specifically on the detected crowd regions. Furthermore, a crowd density estimation algorithm can also be utilized to count non-human objects, such as cells [1] or vehicles [2].

A crowd density estimation algorithm mainly targets congested scenes, such as the images shown in Fig. 5.1. In a congested scene, many people are occluded by others. Furthermore, when a crowd is located at a far distance from the camera, each person may only be represented by a few pixels in an image. Due to challenging issues like occlusion and a small occupied region by individuals in a congested scene, it is hard to count the exact number of people in a crowd. Unlike early detection-based methods that counted individuals one by one, regression-based density estimation methods can efficiently learn a crowd density map by using only point annotations that mark the

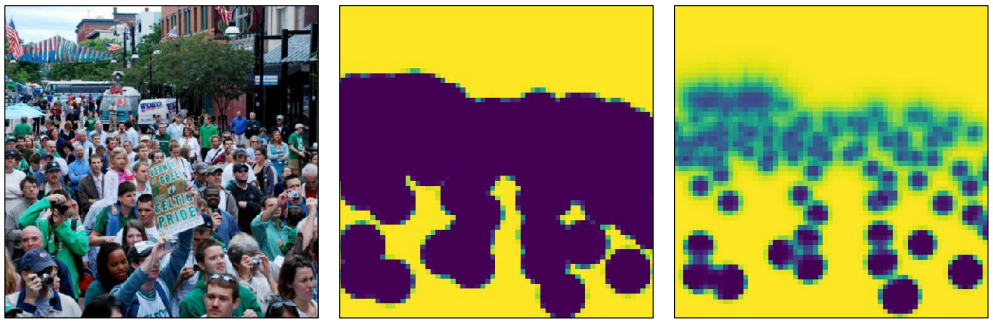
(a) Unevenly distributed crowds (b) Variability in pedestrian size

Figure 5.1: Examples of target scenes for crowd counting. Crowd counting algorithms mainly target highly congested scenes with (a) an unevenly distributed crowd and (b) a distribution of pedestrians of various scales.

location of each person in the image [1, 13, 14].

Regression-based methods have show a large improvement with the advancement of deep learning [12]. Among the deep learning-based methods, the Bayesian loss (BL) method [38] shows impressive performance in training a deep network for crowd density map estimation. Instead of the conventional method that evaluates loss with a desired density value at each pixel, the BL method adopts a novel loss scheme using only the positions of the head point annotations. In contrast to providing the desired density map in conventional methods, the BL method uses a probability that each pixel belongs to a person or background.

In the BL method, the background probability at a pixel is generated by using a fixed distance between the pixel and the nearest head point annotation. However, due to the fixed distance, this background probability model cannot adapt itself to the variation of personal scales and the sparsity of individuals, as shown in Fig. 5.2. The issue mentioned above results in the limited performance for the various sizes of people from a few pixels to a full face or more, which depends on the scale of person. In addition, the BL method cannot handle the varying degrees of occlusion that arise due to the different sparsity of individuals in a certain region.



(a) Image (b) Bayesian Loss method (c) Ours

Figure 5.2: Comparison of background probability map from Bayesian Loss (BL) method and the proposed method. Given a crowd image of (a), the yellow-colored region in (b) and (c) represents the background region indicated by the background probability.

In this chapter, to solve the issue above, we propose a *congestion-aware Bayesian loss* method in which the estimated scale is used to set up a background probability that is adaptable to personal scale variations. To this end, we have developed schemes to estimate the scale of each person and the sparsity of a local region. These schemes are designed under the assumption that the scale of a person is inversely proportional to the distance the individual and the camera, whereas the sparsity of a region is related to the ratio of the scale and the inter-person distance of the region. Unlike the existing scale-aware schemes [4, 9, 25, 30], the proposed scale inference method targets the situation where only point annotations are given. Therefore, our method is suitable for single-image crowd density estimation algorithms that provide training images and corresponding point annotations. The estimated sparsity is used to reduce or amplify the loss to adjust for the difficulty in heavily occluded regions. By using the proposed loss, we can learn a diversity of crowd appearances in a weakly supervised manner with only head point annotations instead of density map annotations. Because a diversity of appearances dependent on scale and sparsity are learned in the training phase, estimations of scale and sparsity are not needed at all in the testing phase, and

therefore, additional inference costs are not accrued.

Through various experiments, we validate the proposed components including the scale and sparsity estimations of the BL, which contribute to the performance improvement of the proposed method in achieving the state of the art with various benchmark datasets.

Contributions of this chapter are summarized as follows:

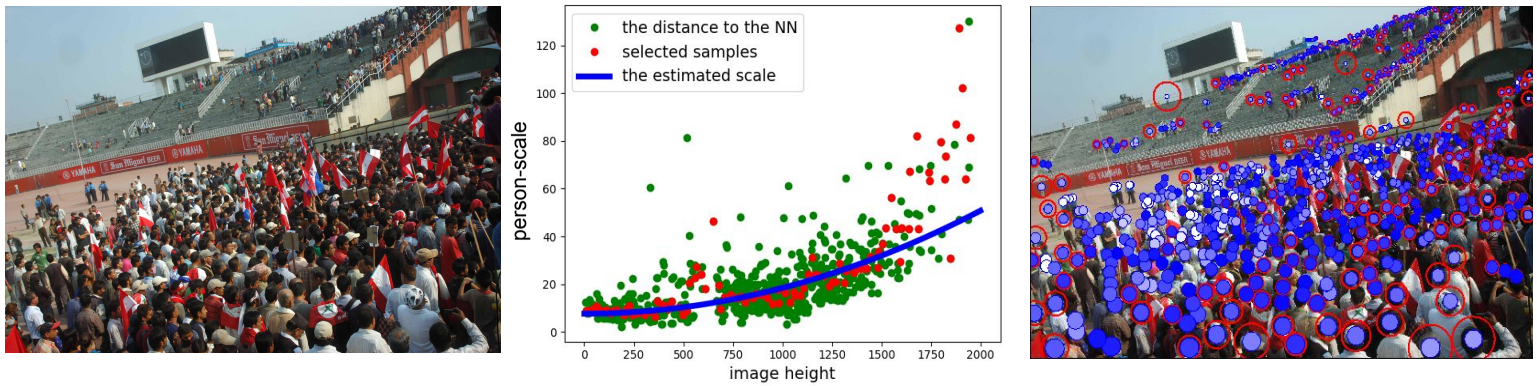
- We develop schemes to estimate the scale of each person (*i.e.*, person-scale) and the sparsity of a local crowd (*i.e.*, crowd-sparsity) based on the scene geometry.
- Using the estimated person-scale and crowd-sparsity, we propose an extended Bayesian loss method to learn a variety of appearances in a crowd.
- Using the proposed Bayesian loss method, we improve the supervising representation of the point annotations and achieve state-of-the-art performance.

5.2 Congestion-aware Bayesian Loss

In this section, we present the estimation procedures of the scale of a person (*i.e.,* person-scale) and the sparsity of a local region (*i.e.,* crowd-sparsity) and then describe the proposed loss using the estimated person-scale and crowd-sparsity. First, the person-scale estimation procedure is described in Sec. 5.2.1. The method for crowd-sparsity estimation is then described in Sec. 5.2.2. With the estimated person-scale and crowd-sparsity, the proposed loss is described in Sec. 5.2.3.

5.2.1 Person-Scale Estimation

To estimate a person-scale in an image, we use the following two scene characteristics. First, the person-scale is represented as inversely proportional to the distance from the person to the camera. We assume a typical surveillance situation where only one ground plane exists, such as a scene without additional layers. In that situation, every person



(a) Input image (b) Fitting of scale trend (c) Scale estimation result

Figure 5.3: The scale estimation procedure of the proposed method. Given a crowd image (a), the distribution between the nearest neighbor distance and image height of annotations is shown in (b). With the nearest neighbor distances (*green points*), we fit a line (*blue line*) using random sample consensus (RANSAC) by sampling (*red points*) evenly within each section. The circles in (c) express the nearest neighbor distance (*red*) and the estimated scale of the crowd after the fitting process (*blue*). If the estimated scale fits the nearest neighbor distance, it is colored blue; otherwise, it is colored white.

at the same image height is assumed to have the same scale. Also, the person-scale is proportional to the image height that is generally defined in ascending order from the top to the bottom of the image. That is, as shown in Fig. 5.3(a), people in the bottom region of an image are represented in a large scale, and *vice versa*. Second, in a congested scene, where people are distributed evenly, the person-scale is represented by the nearest neighbor distance of each person..

Under the assumptions described above, we can estimate the person-scale $s(h)$ at the image height h using the inter-person distance as

$$s(h) \approx \frac{1}{P} \sum_{i=1}^{P} |p_i - p_{\mathcal{N}(i)}|, \tag{5.1}$$

where P is the number of head points at the image height h, p_i is the head position of the i-th person, and $p_{\mathcal{N}(i)}$ is the head position of the nearest neighbor of the i-th person.

However, in some cases, if we directly estimate the person-scale using Eq. (5.1), the scale estimation results can be noisy because outliers can exist with sparsely distributed people. To resolve the outlier issue, we use a regression of the height-scale relationship, as depicted in Fig. 5.3. From Fig. 5.3(b), it can be observed that our assumption on the relation between person-scale and image height is valid. To fit the relationship between person-scale and the image height, we (1) follow the aforementioned scene geometry and (2) consider unevenly distributed crowds as outliers. Hence, we conduct a second-order linear RANSAC (random sample consensus) operation without fitting a constant of the first-order variable, in other words, find a and b in $ax^2 + b$ such that most of the points of x are satisfied. The fitted curve in Fig. 5.3(b) for estimating the person-scale is obtained by the RANSAC regressor, which models the observed data with little influence of outliers. We utilize the estimated person-scale in designing the congestion-aware Bayesian loss method in Sec. 5.2.3.

5.2.2 Crowd-Sparsity Estimation

We can utilize the estimated crowd-sparsity to improve the learning capability of the crowd density estimation network. When learning the crowd density map, in a densely crowded region, it is difficult to distinguish the crowd from the background clutter. Thus, regions of low crowd-sparsity will significantly affect the overall counting performance. Motivated by the hard-negative mining in object detection algorithms [11], we reduce the influence of loss on annotations in sparsely crowded regions, and amplify the influence of loss on annotations in densely crowded regions.

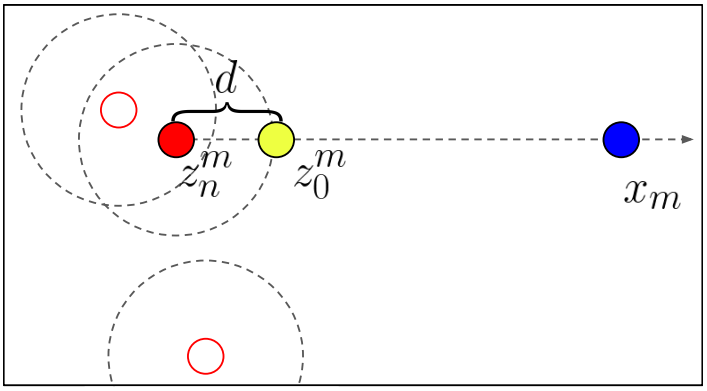
To estimate crowd-sparsity, we utilize the estimated person-scale in Eq. (5.1). If a person has a greater distance to his/her nearest neighbor than the estimated scale, we can assume that the crowd in a local region around the person is sparsely distributed, and *vice versa*. The crowd-sparsity around a person is then defined by the ratio of the nearest neighbor distance of the person to the estimated person-scale, in other words,

$$S_n = s(h_n)/s'(h_n), \tag{5.2}$$

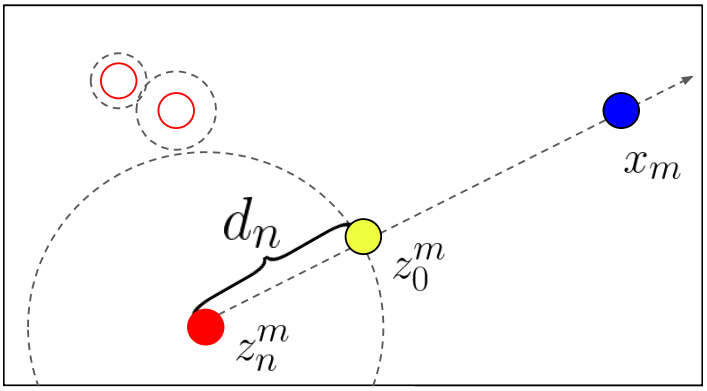
where h_n is the image height of the n-th person, $s(h_n)$ is the distance to his/her nearest neighbor given by $|p_n - p_{\mathcal{N}(n)}|$, and $s'(h_n)$ is the estimated scale for the person. In the region under the fitted curve in Fig. 5.3(b), the people are highly occluded, so S_n is less than one. In contrast, the people in the region above the curve are sparsely distributed, so S_n becomes larger than one. Hence, using the crowd-sparsity S_n, we can reduce or amplify the influence of the annotations depending on the crowd-sparsity of a local region. In Sec. 5.2.3, we describe the derivation of the proposed loss including the estimated person-scale and crowd-sparsity.

5.2.3 Design of The Proposed Loss

Let x_m $(m = 1, 2, ..., M)$ be a random variable that denotes the spatial location, where M is the number of pixels. Given N number of people, z_n $(n = 1, 2, ..., N)$ is a head point annotation. The label for z_n is defined by a random variable y_n. Assuming that the



(a) Bayesian Loss method



(b) Ours

Figure 5.4: Dummy background annotation settings. For a pixel x_m of a density map, (a) the Bayesian loss method adopts a dummy background annotation, z_0^m, at a distance d pixels from z_n^m, which is the nearest neighbor head annotation of x_m. In contrast, (b) the proposed method adopts an adaptable distance d_n depending on the person-scale instead of the fixed d.

likelihood of the head point annotation follows a Gaussian distribution, the likelihood probability of x_m given label y_n is given by

$$p(x_m|y_n) = \mathcal{N}(z_n, \sigma^2 \mathbb{I}_{2\times 2}), \tag{5.3}$$

where σ is a parameter that controls a region that is affected by each head point annotation and $\mathbb{I}_{2\times 2}$ denotes an identity matrix. In addition, given background label y_0, the likelihood probability of x_m is set to a Gaussian kernel with a centroid z_0^m as

$$p(x_m|y_0) = \mathcal{N}(z_0^m, \sigma^2 \mathbb{I}_{2\times 2}). \tag{5.4}$$

In this chapter, in contrast to the original Bayesian loss, we propose an adjustable centroid z_0^m depending on the person-scale. As shown in Fig. 5.4, in the original work [38], the centroid z_0^m is located at a distance d pixels from the nearest head annotation. In our method, the adjusted centroid z_0^m is located at a distance d_n pixels from the nearest head annotation, where d_n depends on the person-scale $s'(h_n)$. To this end, we define d_n by

$$d_n = d_0 \cdot s_I \cdot exp(\frac{s'(h_n) - s_0}{s_0}), \tag{5.5}$$

where s_I is the shorter side length of the image, d_0 (*e.g.*, 0.15) is a fractional scale of s_I, and s_0 is the average person-scale of the dataset. When the person-scale $s'(h_n)$ becomes larger than the average scale s_0, we set d_n to grow exponentially. The adjusted centroid is then obtained by

$$z_0^m = z_n^m + d_n \frac{x_m - z_n^m}{\|x_m - z_n^m\|_2}, \tag{5.6}$$

where z_n^m denotes the nearest annotation point of x_m.

Using the likelihoods, given the spatial position x_m, the posterior probability of each head point annotation or background is given by

$$p(y_n|x_m) = \frac{p(y_n)p(x_m|y_n)}{\sum_{n'=0}^{N}[p(y_{n'})p(x_m|y_{n'})]}, \tag{5.7}$$

where $p(y_n) = \frac{1}{N+1}$ denotes the prior probability with label index $n = 0, 1, 2, ..., N$, including the background.

If the posterior probability of each head point annotation in Eq. (5.7) is expressed as a map, it represents the contributed region of each head annotation. Similarly, the posterior probability for the background annotation can represent the background region, as illustrated in Fig. 5.2(c). It can be seen that the proposed method more accurately represents the background according to the person-scale of annotations than the original work in Fig. 5.2(b).

If an estimated crowd density at location x_m is denoted as $D^{est}(x_m)$, the Bayesian loss is derived as follows. Let c_n^m be a count at x_m contributed by y_n, and c_n is a count of n-th annotation. Following [38], the expectation of c_n is derived as

$$\begin{aligned} E[c_n] &= E[\sum_{m=1}^{M} c_n^m] = \sum_{m=1}^{M} E[c_n^m] \\ &= \sum_{m=1}^{M} p(y_n|x_m) D^{est}(x_m). \end{aligned} \tag{5.8}$$

The count value of each annotation c_n should be one and that of background c_0 should be zero. Using the crowd-sparsity for each annotation in Eq. (5.2), the proposed congestion-aware Bayesian loss (CBL) is proposed by

$$\begin{aligned} \mathcal{L}_{CBL} &= \sum_{n=1}^{N} \frac{1}{S_n} |1 - E[c_n]| + |E[c_0]| \\ &= \sum_{n=1}^{N} \frac{1}{S_n} \left| 1 - \sum_{m=1}^{M} p(y_n|x_m) D^{est}(x_m) \right| \\ &\quad + \sum_{m=1}^{M} p(y_0|x_m) D^{est}(x_m), \end{aligned} \tag{5.9}$$

where S_n reduces or amplifies the influence of the annotations depending on the crowd-sparsity of a local region. At inference time, we can obtain the number of people without

Table 5.1: Network structure. The configuration of the convolution layer is expressed as [kernel size]-[number of channels].

Layer	Feature Extraction	Layer	Regression
1-1,2	conv3-64		Bilinear Interpolation
	max pool	1	conv3-256
2-1,2	conv3-128	2	conv3-128
	max pool	3	conv3-1
3-1,2,3,4	conv3-256		
	max pool		
4-1,2,3,4	conv3-512		
	max pool		
5-1,2,3,4	conv3-512		

the posterior label probability $p(y_n|x_m)$ as follows:

$$\begin{aligned} C &= \sum_{n=1}^{N} E[c_n] = \sum_{n=1}^{N}\sum_{m=1}^{M} p(y_n|x_m) D^{est}(x_m) \\ &= \sum_{m=1}^{M}\sum_{n=1}^{N} p(y_n|x_m) D^{est}(x_m) \\ &= \sum_{m=1}^{M} D^{est}(x_m), \end{aligned} \tag{5.10}$$

where C denotes the number of people in the entire image.

5.3 Experiments

In this section, we describe the evaluation of the effectiveness of the proposed components and illustrate that our method was able to achieve the state-of-the-art on various benchmark datasets.

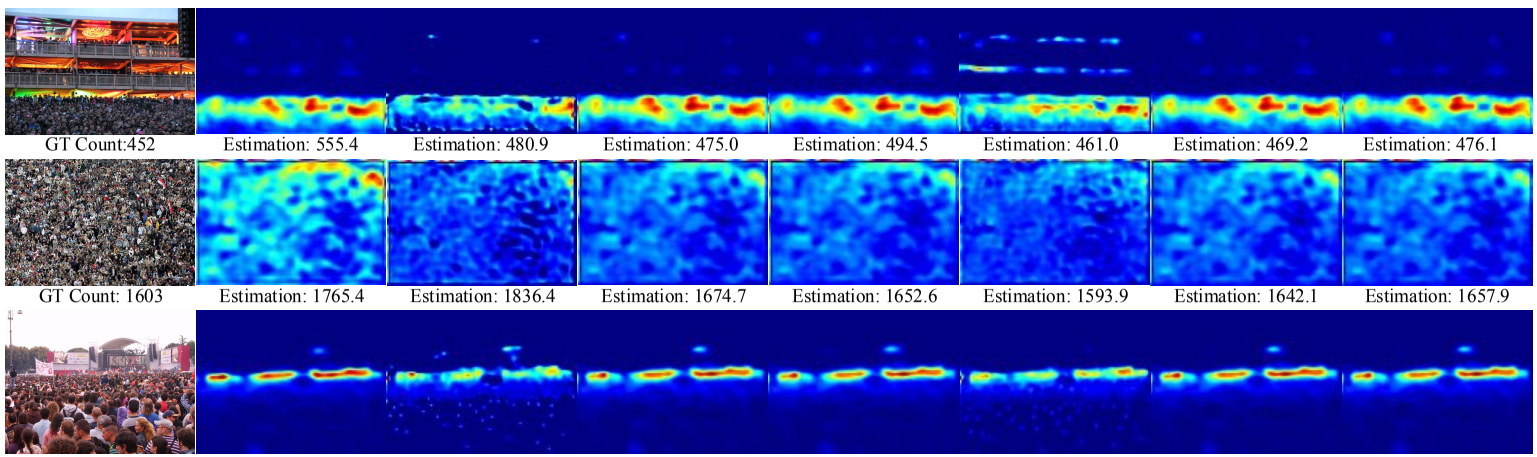


Figure 5.5: Qualitative results of the ablation study: (a) Input image; (b) Estimated density maps from the baseline; (c) Estimated density maps after the scale estimation; (d–h) Estimated density maps after sparsity estimation, varying the fraction of the shorter side of the input image, (d) $d_0 = 1.5$, (e) $d_0 = 2$, (f) $d_0 = 2.25$, (g) $d_0 = 2.5$, and (h) $d_0 = 3$.

Table 5.2: Datasets for experiments

Dataset	Images	Mean Resolution	Annotations
UCF_QNRF	1,535	2338×1607	1,007,316
ShanghaiTech Part A	482	873×599	162,413
ShanghaiTech Part B	716	1024×768	49,151
UCF_CC_50	50	2101×2888	63,974

5.3.1 Datasets

As summarized in Table 5.2, we have evaluated the proposed method on four challenging crowd counting datasets: UCF_QNRF, ShanghaiTech Part A, ShanghaiTech Part B, and UCF_CC_50.

• **UCF_QNRF [70]** is the latest and largest crowd counting dataset, which includes 1,535 images crawled from Flickr with 1.01 million point annotations. It is a challenging dataset because it has a wide range of counts, image resolutions, light conditions, and viewpoints. The training set has 1,201 images and the remaining 334 images are used for testing.

• **ShanghaiTech [5]** contains 1,198 images with a total of 330,165 people and is divided into two parts: Part A containing 482 images of congested scenes (300 images for training and 182 images for testing), and Part B containing 716 images of sparse scenes (400 images for training and 316 for testing).

• **UCF_CC_50 [70]** contains only 50 gray-scale images which are considered to be challenging due to the high crowd density in the images. Its count value varies from 94 to 4,543. Due to its small quantity, experiments are conducted by 5-fold cross validation followed by the original literature [70].

5.3.2 Implementation Details

The proposed network consists of a VGG19 CNN model as described in Table 5.1. We trained the network in an end-to-end fashion. The first 19 convolutional layers were initialized with a pre-trained VGG19. For the data augmentation processes, we performed random flipping and the cropping of the given images with a size of 512×512 for the UCF_QNRF, ShanghaiTech Part A and UCF_CC_50 datasets and 256×256 for the ShanghaiTech Part B dataset. The parameters were updated by an Adam (adaptive moment estimation) optimizer. All the experiments were performed on an NVIDIA 1080Ti GPU.

5.3.3 Evaluation Metrics

To evaluate our proposed method, we used both the mean absolute error (MAE) and mean squared error (MSE) as evaluation metrics:

$$MAE = \frac{1}{L}\sum_{i=1}^{L}\left|C_i - C_i'\right|, \tag{5.11}$$

$$MSE = \sqrt{\frac{1}{L}\sum_{i=1}^{L}|C_i - C_i'|^2}, \tag{5.12}$$

where L is the number of test images, C_i is the number of people in the i-th image, and C_i' is the estimated number of C_i. The number of people in the image is obtained by the integration of the crowd density over all the image regions as

$$C_i = \sum_{m=1}^{M} D^{GT}(x_m). \tag{5.13}$$

Similar to the approach used in Eq. (5.13), the estimated count is obtained as follows:

$$C_i' = \sum_{m=1}^{M} D^{est}(x_m). \tag{5.14}$$

Table 5.3: Experimental results for ablation study

	Base	Scale	Sparsity with varying d_0				
			1.5	2	2.25	2.5	3
MAE	68.7	73.6	68.5	68.3	**61.8**	65.7	66.4
MSE	114.7	115.5	105.3	104.0	**101.7**	103.1	102.8

5.3.4 Ablation Study

In this section, we describe the conduct of several experiments to verify the extent to which each proposed component contributed to performance improvement. The ablation experiments were performed on the ShanghaiTech Part A dataset because it could represent well the effectiveness of the proposed method due to its diversity of person-scale and crowd-sparsity within a relatively small quantity of images. According to the configuration of the proposed method, the following three cases were tested:

- **Base** was conducted the same way of Bayesian loss [38]. d_0 in Eq. (5.6) was set to 0.15.
- **Scale** had the same setting as **Base**, including the proposed person-scale estimation process.
- **Sparsity** trains the network with the proposed loss, including the proposed crowd-sparsity estimation in Eq. (5.9).

The proposed method has only one hyper-parameter, d_0, which is a guideline for estimating the person-scale. If d_0 varies, the represented scale also varies as the proposed definition. Therefore, we also conducted a comparison experiments varying d_0 after adopting the **Sparsity** setting from the ablation study, which was named **Sparsity-**d_0.

The qualitative results of the ablation study are depicted in Fig. 5.5, and the quantitative results are summarized in Table 5.3. Among the testing cases, the best performance

was achieved when d_0 was set to 2.25 while considering both the person-scale and crowd-sparsity. The following analysis was derived from the ablation study.

- ***Scale*** *improved the representation of individual's locations but slightly lost counting accuracy when compared with* ***Base***. As shown in Figs. 5.5(a) and (c), estimated density map of a large-scaled person in the front is represented by a point-shape in **Scale** (c). The point-shaped result means that the density estimation network accurately estimated the location of a person; however, it resulted in a slight loss of some of the counting performance. A strict restriction of point annotation could lead to an inaccurate estimation of the density map around a person. As depicted in Fig. 5.2(c), performing person-scale estimation concentrates more on the location of the head point annotation than the original work. It could falsely learn the density in more tightly crowded regions containing noisy annotations.

- ***Sparsity-1.5*** *started to improve performance compared to* ***Base***. When d_0 in Eq. (5.6) was set to be 10 times larger (*i.e.*, $d_0 = 1.5$) than the original work, the performance became similar. In other words, when d_0 was set to 1.5, the training started to consider the diversity of the person-scale without losing the counting performance. As shown in the first row of Figs. 5.5(c) and (d), false positives were reduced at the top of the estimated map in **Sparsity-1.5** (d), compared to **Scale** (c). Also, **Sparsity-1.5** successfully estimated the density at the bottom of the first row of (d), which was incorrectly estimated as zero in (c) by **Scale**.

- ***Sparsity-2.25*** *showed the best performance.* As depicted in the first row of Fig. 5.5, a small-scaled individual at the bottom of the image was hard to represent in the density map, except for **Sparsity-2.25**. We can observe the effect of the proposed method through the third row of Fig. 5.5(f) in which the density in the background region was successfully estimated to be zero; in the other cases, false positives were shown in the background region.

- ***Sparsity*** *settings except for* $d_0 = 2.25$ *had similar density map representations.* We can see that every setting except for **Sparsity-2.25** failed to learn the hard cases mentioned above, such as missing small-scaled people and the falsely estimated densities in the background region. From the results, we confirm that only one parameter setting (d_0 = 2.25) improved both counting accuracy and representation capability.

In the remaining experiments, d_0 was set to 2.25 for all the datasets according to the results from the ablation study.

5.3.5 Comparisons with State of the Art

For the four datasets (UCF_QNRF, ShanghaiTech Part A, ShanghaiTech Part B and UCF_CC_50), we performed extensive comparison experiments with 16 state-of-the-art algorithms, including the early deep-learning models (CCNN [4] and MCNN [5]), models with novel network structures (CMTL [36], SCNN [19], CP-CNN [23], AC-SCP [22], DCNet [74], IG-CNN [27], IC-CNN [73] , CL-CNN [75], DA-Net [75], ISANet [76], and SDSP [26]), models with network layers specialized in the crowd density estimation (SANet [28], SAAN [25], CSRNet [21] and CAN [29]), a model based on detection scheme [10], and BL method [38]. As summarized in Table 5.4, the proposed method CBL exhibited the best performance on the MAE metric and also showed a competitive result on the MSE metric. A noticeable improvement was found in the UCF_QNRF, ShanghaiTech Part A, and UCF_CC_50 datasets, in which at least thousands of people were depicted in images. In contrast, it was limited in finding a performance improvement in the ShanghaiTech Part B dataset, which consisted of hundreds of people in relatively simple surveillance environments with few occlusions.
• **UCF_QNRF**: Fig. 5.6 illustrates the qualitative results for UCF_QNRF dataset. In the first column, false positives in the background were removed more in our method compared to the BL method. It was because the foreground and the background were well separated by the proposed person-scale estimation. In the second and the fourth column,

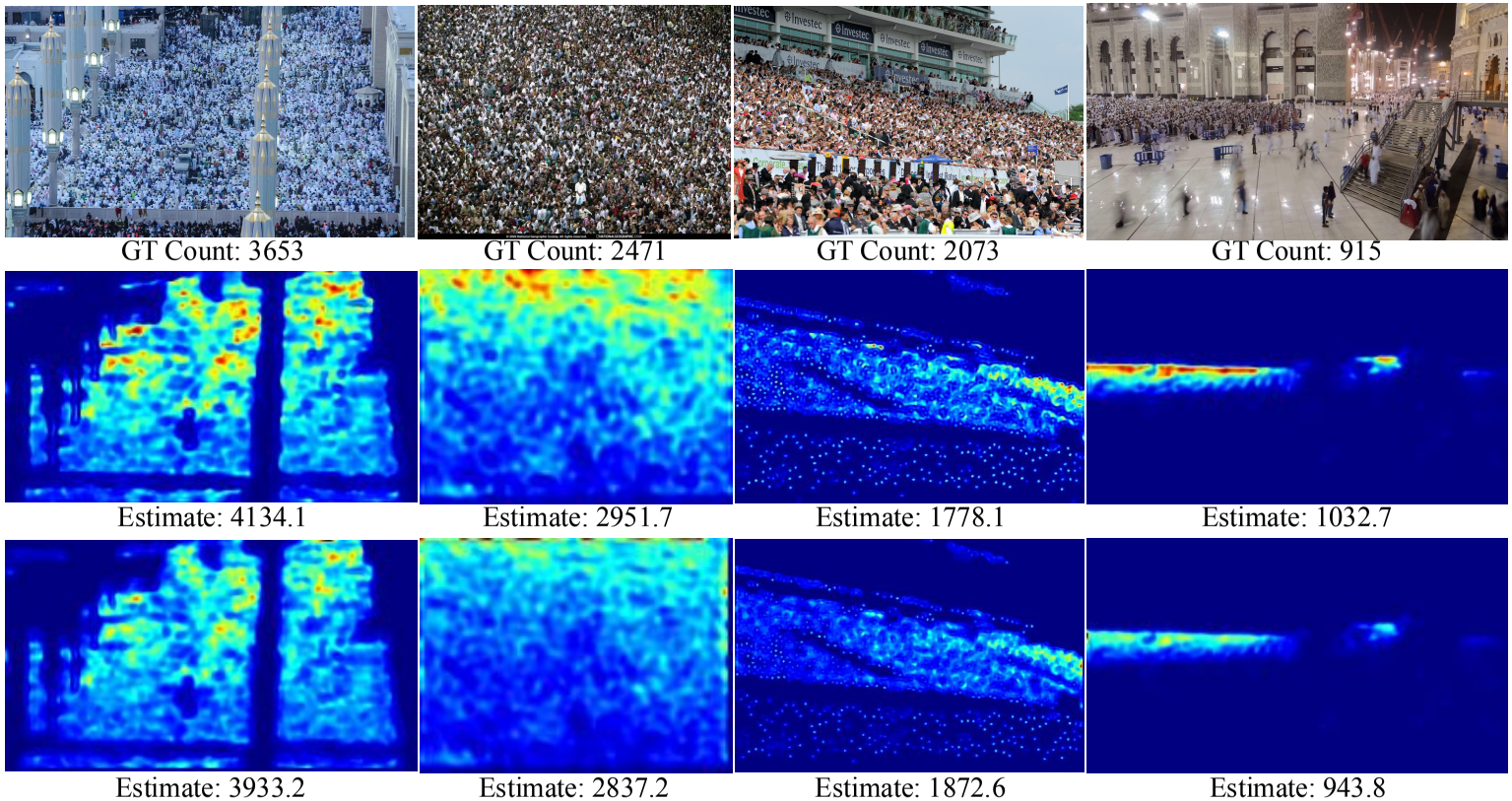


Figure 5.6: Qualitative results with the UCF_QNRF. *Top row*: Input image; *Middle row*: Bayesian loss [38]; *Bottom row*: Our method.

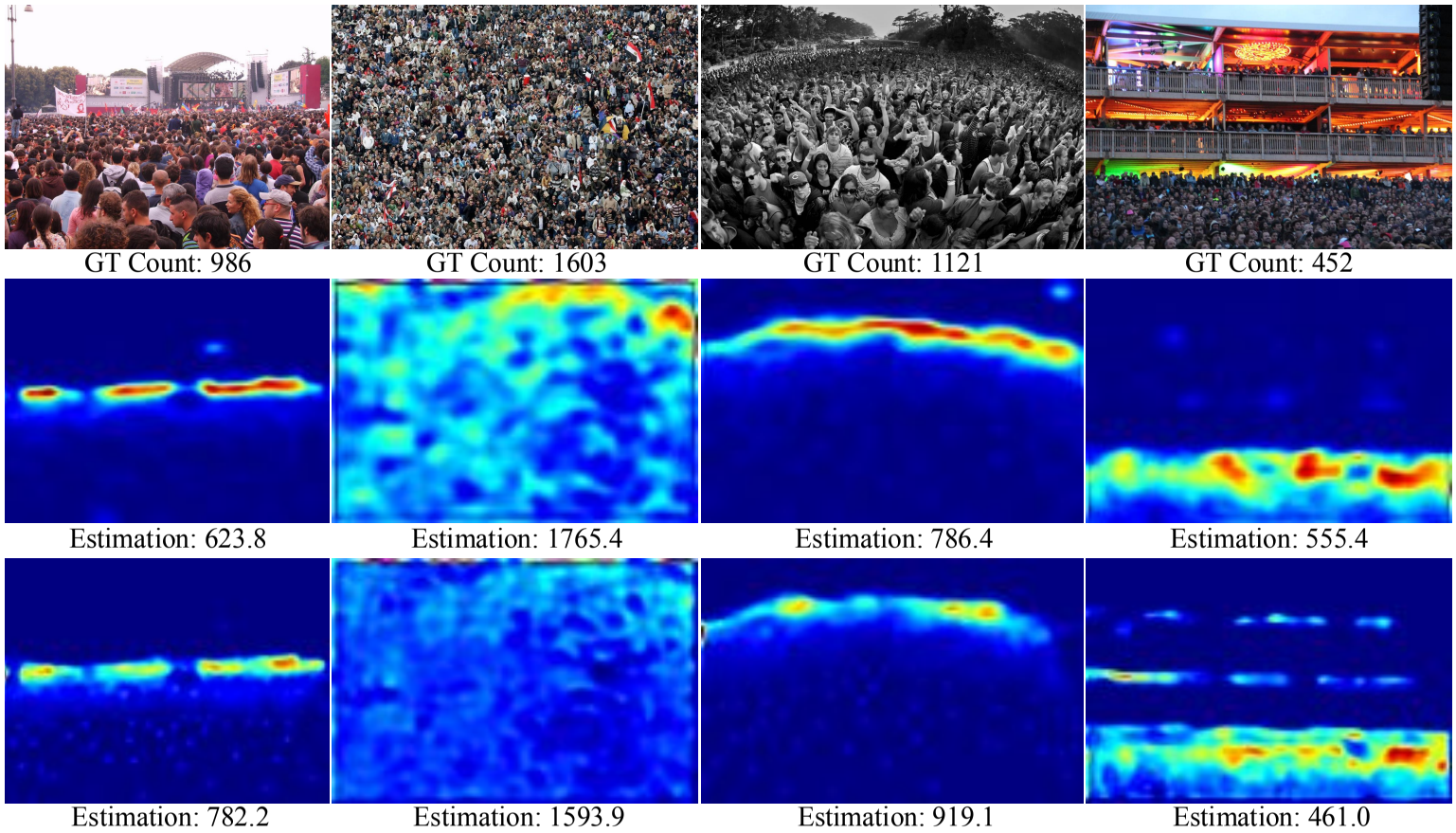


Figure 5.7: Qualitative results with ShanghaiTech Part A. *Top row*: Input image; *Middle row*: Bayesian loss [38]; *Bottom row*: Our method.

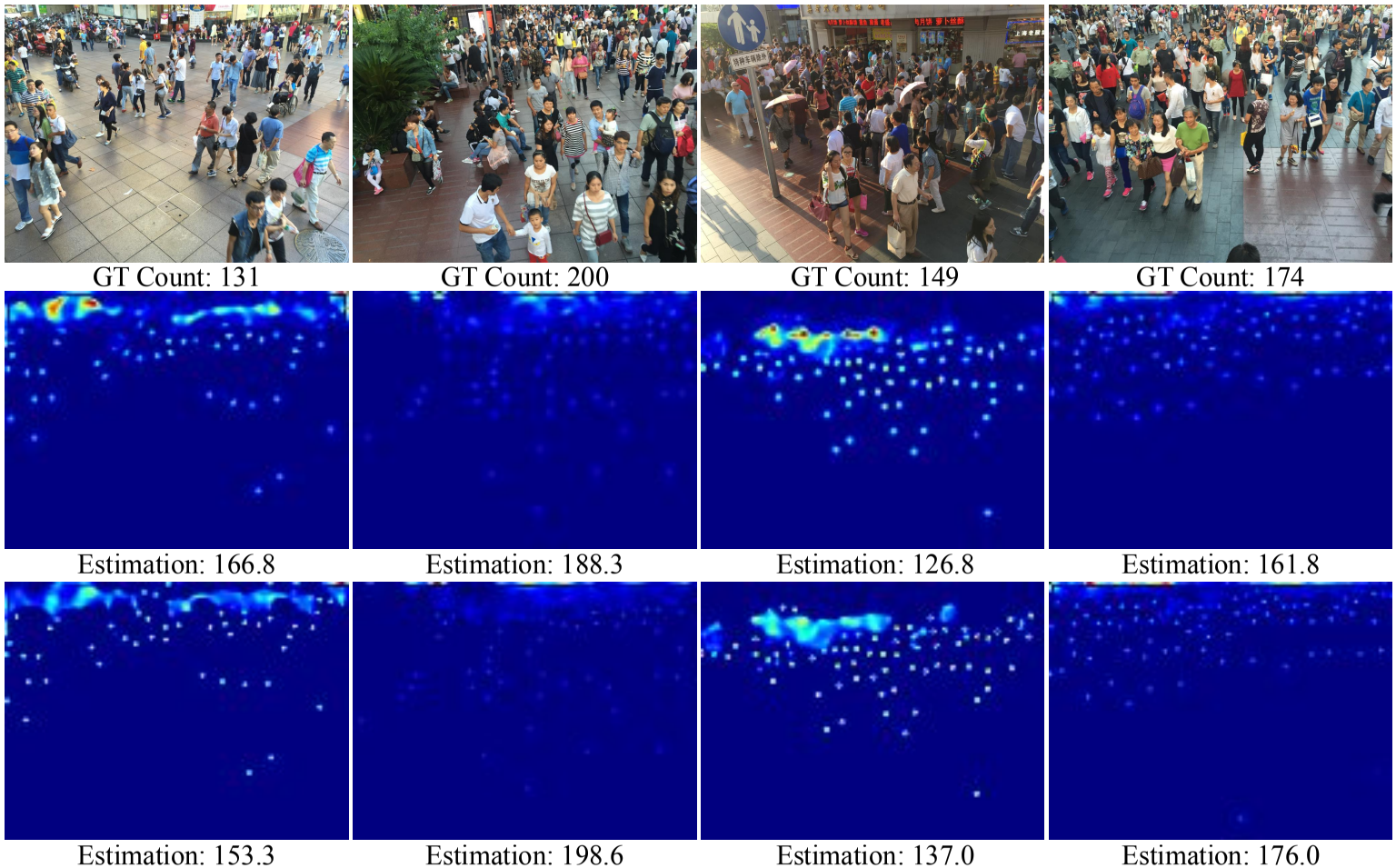


Figure 5.8: Qualitative results with ShanghaiTech Part B. *Top row*: Input image; *Middle row*: Bayesian loss [38]; *Bottom row*: Our method.

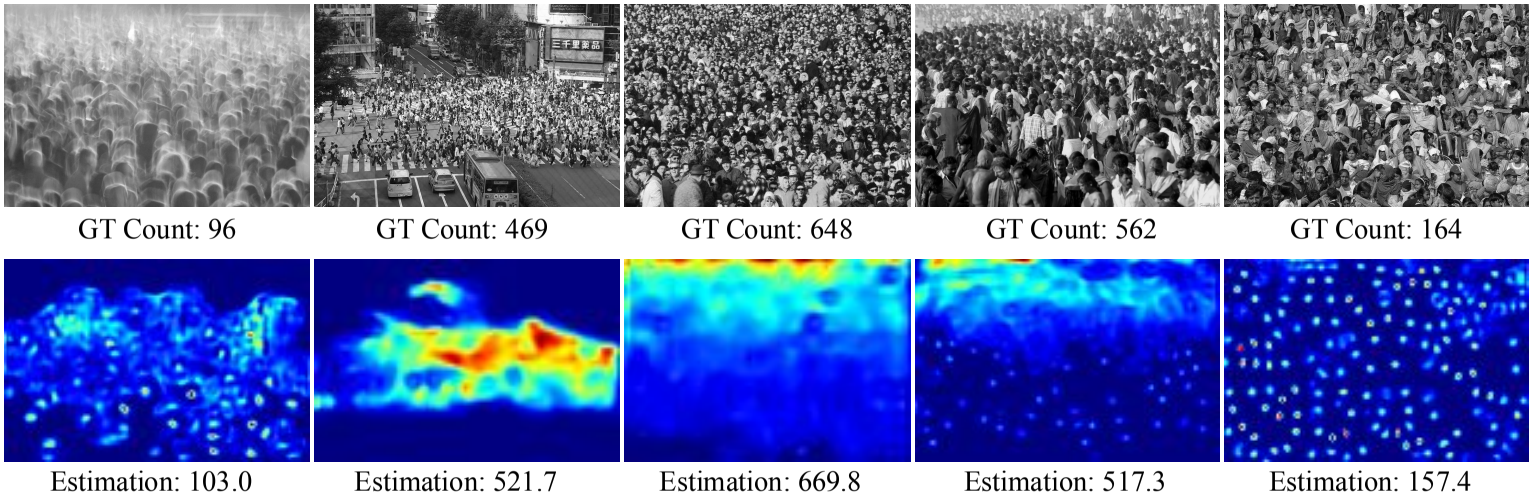


Figure 5.9: Qualitative results with UCF_CC_50. *Top row*: Input image; *Bottom row*: Our method.

Table 5.4: Experimental results for comparison with state-of-the-arts

Method	UCF_QNRF		SHT Part A		SHT Part B		UCF_CC_50	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
CCNN [4]	-	-	181.8	277.7	32.0	49.8	467.0	498.5
MCNN [5]	277	426	110.2	173.2	26.4	41.3	377.6	509.1
CMTL [36]	252	514	101.3	152.4	20.0	31.1	322.8	341.4
SCNN [19]	228	445	90.4	135.0	21.6	33.4	318.1	439.2
CP-CNN [23]	-	-	73.6	106.4	20.1	30.1	295.8	320.9
ACSCP [22]	-	-	75.7	102.7	17.2	27.4	291.0	404.6
DCNet [74]	-	-	73.5	112.3	18.7	26.0	288.4	404.7
IG-CNN [73]	-	-	72.5	118.2	13.6	21.1	291.4	349.4
CSRNet [21]	98.2	157.2	68.2	115.0	10.6	16.0	266.1	397.5
IC-CNN [27]	-	-	68.5	116.2	10.7	16.0	260.9	365.5
SANet [28]	-	-	67.0	104.5	8.4	13.6	258.4	334.9
SAAN [25]	-	-	-	-	16.9	28.4	271.6	391.0
CL-CNN [75]	132	191	-	-	-	-	-	-
DA-Net [76]	-	-	71.6	104.9	15.0	21.9	290.8	326.5
ISANET [77]	-	-	75.8	124.9	11.0	18.6	-	-
CAN [29]	107.0	183	62.3	**100.0**	7.8	**12.2**	212.2	**243.7**
SDIHD [10]	112	173	-	-	-	-	-	-
SDSP [26]	115.2	175.7	-	-	-	-	229.4	325.6
BL [38]	88.7	**154.8**	62.8	101.8	**7.7**	12.7	229.3	308.2
CBL (Proposed)	**87.0**	155.8	**61.8**	101.7	**7.7**	13.1	**191.7**	283.0

the BL method provided an overestimation in a congested region, while such errors were reduced in the proposed method. It is inferred that the propose method provided more accurate learning in the congested region to improve the counting performance. In the third column, however, the localization performance became degraded, resulting in a reduced resolution of crowd density in the grandstand region. It is because people that are densely crowded and severely occluded made the representation in the density map worse.

• **ShanghaiTech Part A**: Fig. 5.7 depicts the qualitative results for the ShanghaiTech Part A dataset. In the first and third column, the representation of the density map was improved in the region where people were sparsely distributed. Density regions that were not counted in the BL method were now expressed in detail, and the underestimated regions were improved. In the second column, which was a highly congested situation, our method more accurately counted the crowd compared to the BL method by improving the crowd's representation and reducing overestimations. In the fourth column, the accuracy was improved from the accurate separation of the foreground and the background. The BL method often failed to count people near the top of the image and on the railing with complex patterns because of the errors made in these regions. Our method accurately recognized not only the people on the railing but also people in the congested region.

• **ShanghaiTech Part B**: Fig. 5.8 shows the qualitative results from the ShanghaiTech Part B dataset. Because the number of people was smaller than with the other datasets and there were few crowded situation, performance improvement with the proposed method was limited. There was no meaningful difference between the BL method and the proposed method. This was because the proposed method learns various surveillance environments, while this dataset had almost the same scale distribution over the sample images. A slight improvement was achieved in partially crowded regions, such as in the first and the third columns of Fig. 5.8. Although the qualitative results looked similar, the counting accuracy was improved for the whole case in the sample images in the

second and fourth column of Fig. 5.8.

• **UCF_CC_50**: Fig. 5.9 shows the qualitative results from the UCF_CC_50 dataset. Since the qualitative results can be slightly different depending on randomly selected samples in the cross-validation setting of the UCF_CC_50 dataset, only the qualitative results of the proposed method are presented. In the second column, the background area is clearly represented by a small density value close to zero. The positions of the people are accurately represented by a point-shape in the last column. In the high-congested scene, such as third column, the estimated density map is blurred, as opposed to the last column, where the estimated density is clearly represented.

5.3.6 Differences from Existing Person-scale Inference

We discuss distinctive aspects of the proposed person-scale inference in contrast to the existing methods as follows.

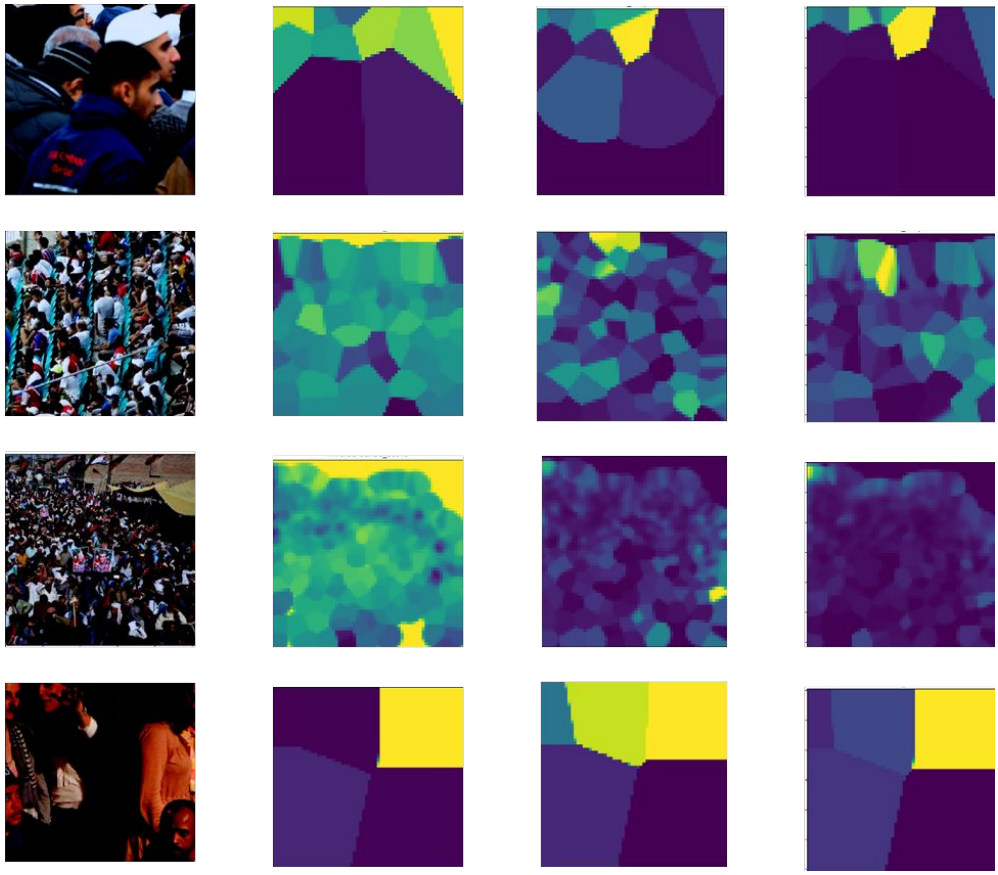
First, there are methods using the built-in person-scale inference module similar to the proposed method. These methods train the networks to infer the scale of a person along with a learning crowd density. In [30], an additional network module is used for data-driven person-scale inference that requires predefined scale-levels for training scale-level-wise branches in the network. Unlike ours, [30] has a limitation that the scale-level must be defined in advance. Also, the additional network module for person-scale inference requires additional computational overhead. In [25], 'scale' is defined by a value inversely proportional to the number of people in a local image patch. In addition, the 'scale' has to be learned as additional feature. Since the 'scale' in [25] is defined under the assumption that people are evenly distributed in the image patch, even the same scale can be measured differently depending on the sparsity of a local region, which leads to inaccurate scale estimation. In contrast, our person-scale estimation is based on the distance from the person to the camera and so the estimation is robust to the sparsity of a local region. Furthermore, [25] requires additional module for learning person-scale, which increases computational overhead.

Second, there are methods to obtain an accurate person-scale using external information such as head detection results [9] or scene perspective information [4]. In [9], the scale inference module is based on head detection results. In this detection-based framework, the person-scale can be accurately inferred in the ideal case, but it is difficult to apply the Bayesian loss framework if humans are falsely detected or undetected. Even if detection performs well, scale inference can depend on the performance of detector to affect crowd density estimation performance. [4] targets scenarios where we provide scene information such as region-of-interest and perspective information. However, in the single-image crowd density estimation settings, scene information is usually not accessible in the training phase.

To sum up, the proposed scale inference method can be applied to various crowd environments. The proposed person-scale inference method enables the scale to be inferred even if a small number of point annotations are given. Also, we consider the limitation of single-image crowd density estimation settings that only the position of the annotation is given.

5.3.7 Analysis on the Proposed Components

An additional experiment was conducted to confirm the effect of crowd-sparsity on the proposed loss. Experiments were performed on the UCF_QNRF dataset. We checked how accurately each person's count was learned for the calculated crowd-sparsity. In the Fig. 5.10, the region with high crowd-sparsity should be trained to reduce the error as much as possible. It can be seen that the area with low crowd-sparsity, that is, the yellow-colored area in Fig. 5.10-(b), is similarly represented to the loss in the proposed method than the existing method. This means that the learning proceeds with a large loss in the regions with low crowd-sparsity as intended.



(a) Image (b) Crowd-Sparsity (c) Error (BL) (d) Error (CBL)

Figure 5.10: **Example figures for in-detail analysis.** Given an input image (a), calculated crowd-sparsities is shown by (b). The corresponding counting errors of both Bayesian loss and our method are shown by (c, d). In all of figures, the bigger value is colored by yellow.

5.4 Summary

In this chapter, we tackled the problem of estimating crowd density accurately on the congested scene for the crowd counting. We proposed a novel congestion-aware loss considering the scale and sparsity of people. The scale of a person (*i.e.* person-scale) was estimated from scene geometry. The sparsity of a local region (*i.e.* crowd-sparsity) was then estimated from the difference between the estimated scale and the nearest neighbor distance. The estimated person-scale and crowd-sparsity was utilized to the proposed congestion-aware loss. We verified the effect of the proposed components through the ablation experiments. From the analysis on the ablation study, the person-scale estimation helped to improve localization accuracy of crowd density, however, degraded the counting performance. We found that the utilizing the crowd-sparsity improves the counting performance while maintaining the localization accuracy. Based on the results from the ablation study, we conducted comparative experiments between the proposed method and the state-of-the-art methods. It was shown that the proposed method showed the state-of-the-art performance. The proposed method showed that the person-scale and crowd-sparsity are important for crowd density estimation. And also, if these two properties are dealt with unified way, we can show that both the counting performance and the localization accuracy could be improved. In future works, additional performance improvement is expected if a unified method is developed.

Chapter 6

Conclusion

In this dissertation, I proposed novel crowd density estimation algorithm in two major directions; network structural perspective and learning strategy perspective.

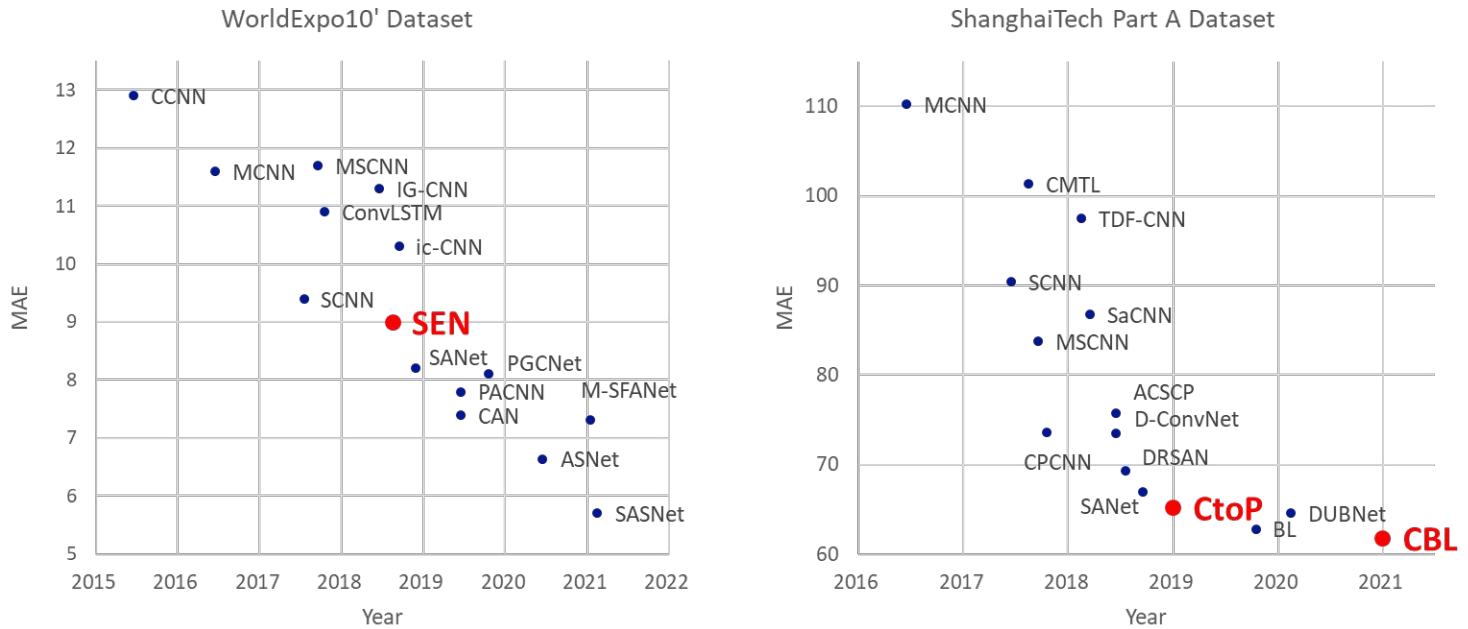
First, I proposed a novel CNN architecture for crowd density estimation that selectively utilizes sub-networks with respect to crowdness. I also propose an adjustable loss scheme for each sub-network that adjusts the balance of counting loss and density loss, depending on crowdness and training epochs. This adjustable loss scheme can also handle the scale issue in which high-density regions are predominantly learned. In addition, the proposed refinement sub-network effectively renders the density map as high resolution map by taking account of contextual information. To the best of our knowledge, this is the first attempt to resolve the trade-off between density map accuracy and counting accuracy by considering both network architecture and loss functions. As the comparative evaluation shows, our network exhibits state-of-the-art performance for three publicly available datasets. The self-evaluation results confirm the validity of the components of the proposed method (selective ensemble, adjustable loss, alternating re-labeling, and refinement sub-network).

Second, I proposed a novel crowd density estimation method which gradually estimates density from the center to periphery of each person. I first showed an empirical finding that the accuracy of the density estimation for the centered or surrounding region

of individuals depends on the scale of dilated convolution. The centered region can be estimated I'll by a small-scaled dilated convolution, while a large-scaled dilated convolution makes small error for the surroundings. Based on the finding, I proposed Cascade Reidual Dilated Network (CRDN) equipped with multiple dilated CNN blocks. CRDN estimates the density of the center of each person with the small-scale dilated CNN block first, and then subsequently estimates the remaining areas with the larger-scale blocks. Extensive experiments show that the proposed method can accurately estimate crowd density over the state-of-the-art.

Third, I proposed improved Bayesian Loss taking into account the estimated scales of crowd and the density level inferred by the estimated scales. In this research, I tackled the problem of accurately estimating crowd density in congested scenes for crowd counting. I proposed a novel congestion-aware loss method that considers the scale and sparsity of people. The scale of a person (*i.e.*, person-scale) was estimated from scene geometry. The sparsity of a local region (*i.e.*, crowd-sparsity) was then estimated from the difference between the estimated scale and the nearest neighbor distance. The estimated person-scale and crowd-sparsity was utilized for the proposed congestion-aware loss. I verified the effect of the proposed components through ablation experiments. From the analysis of the ablation study, the person-scale estimation helped to improve the localization accuracy of the crowd density; however, it degraded the counting performance. I found that utilizing the crowd-sparsity improved the counting performance while maintaining the localization accuracy. Based on the results from the ablation study, I conducted comparative experiments between the proposed method and the state-of-the-art methods. It was shown that the proposed method also demonstrated the state-of-the-art performance. The proposed method illustrated that the person-scale and crowd-sparsity were important for crowd density estimation.

If these two properties were dealt with in a unified way, I could show that both the counting performance and the localization accuracy could be improved. In future works, additional performance improvement is expected if a unified method is developed.



(a) WorldExpo10' dataset (b) ShanghaiTech Part A dataset

Figure 6.1: **Overall performace comparison on Worldexpo10' and ShanghaiTech Part A datasets.** The proposed researches of the dissertation are denoted red dots as; *SEN*: Selective Ensemble Network in Chapter. 3, *CtoP*: Center to Periphery Method in Chapter. 4, and *CBL*: Congestion-aware Bayesian Loss in Chapter. 5.

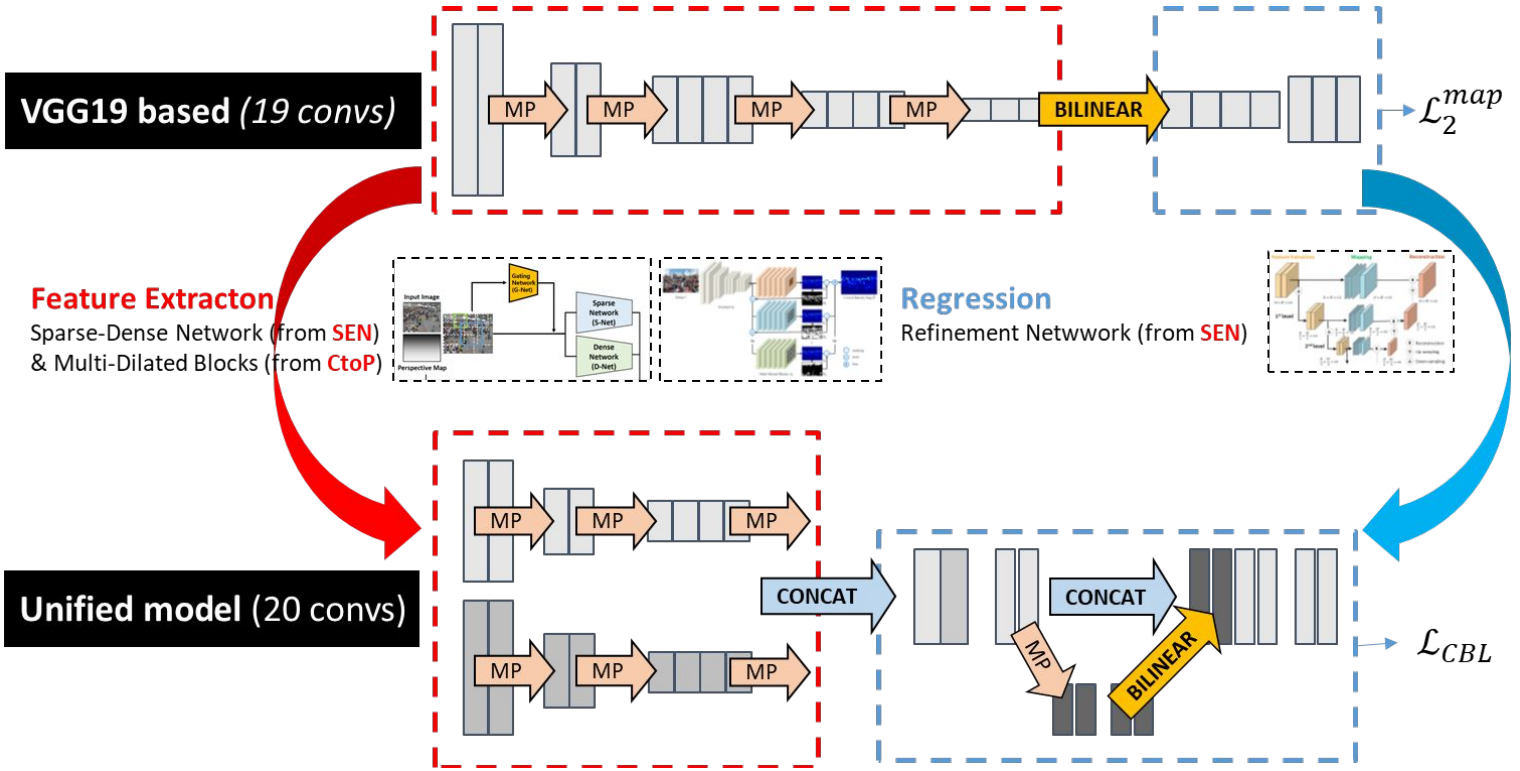


Figure 6.2: **Unified model as a proposal of future work.** Researches in this dissertation are target to either design of congestion-scale-aware network structure or design of congestion-scale-aware training strategy. As a proposal of future work, we designed unified model combining designing themes of proposed researches.

Bibliography

[1] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *BMVC*, volume 1, page 3, 2012.

[2] D. Onoro-Rubio and R. J. López-Sastre. Towards perspective-free object counting with deep learning. In *ECCV*, pages 615–629, 2016.

[3] Geoffrey French, Mark Fisher, Michal Mackiewicz, and Coby Needle. Convolutional neural networks for counting fish in fisheries surveillance video. 2015.

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

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

[6] Chen Change Loy, Ke Chen, Shaogang Gong, and Tao Xiang. Crowd counting and profiling: Methodology and evaluation. In *Modeling, Simulation and Visual Analysis of Crowds*, pages 347–382. Springer, 2013.

[7] Rafael Munoz-Salinas, Eugenio Aguirre, and Miguel García-Silvente. People detection and tracking using stereo vision and color. *Image and Vision Computing*, 25(6):995–1007, 2007.

[8] Min Li, Zhaoxiang Zhang, Kaiqi Huang, and Tieniu Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-

shoulder detection. In *2008 19th international conference on pattern recognition*, pages 1–4. IEEE, 2008.

[9] Sultan Daud Khan, Habib Ullah, Mohammad Uzair, Mohib Ullah, Rehan Ullah, and Faouzi Alaya Cheikh. Disam: Density independent and scale aware model for crowd counting and localization. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4474–4478. IEEE, 2019.

[10] Sultan Daud Khan and Saleh Basalamah. Scale and density invariant head detection deep model for crowd counting in pedestrian crowds. *The Visual Computer*, pages 1–11, 2020.

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[12] V. A. Sindagi and V. M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 2017.

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

[14] Ke Chen, Shaogang Gong, Tao Xiang, and Chen Change Loy. Cumulative attribute space for age and crowd density estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2467–2474, 2013.

[15] V. Lempitsky and A. Zisserman. Learning to count objects in images. In *NIPS*, pages 1324–1332, 2010.

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

[17] Yi Wang and Yuexian Zou. Fast visual object counting via example-based density estimation. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3653–3657. IEEE, 2016.

[18] C. Shang, H. Ai, and B. Bai. End-to-end crowd counting via joint learning local and global count. In *ICIP*, pages 1215–1219, 2016.

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

[20] F. Xiong, X. Shi, and D.-Y. Yeung. Spatiotemporal modeling for crowd counting in videos. *arXiv preprint arXiv:1707.07890*, 2017.

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

[22] Zan Shen, Yi Xu, Bingbing Ni, Minsi Wang, Jianguo Hu, and Xiaokang Yang. Crowd counting via adversarial cross-scale consistency pursuit. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5245–5254, 2018.

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

[24] Jiyeoup Jeong, Hawook Jeong, Jongin Lim, Jongwon Choi, Sangdoo Yun, and Jin Young Choi. Selective ensemble network for accurate crowd density estimation. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 320–325. IEEE, 2018.

[25] Mohammad Hossain, Mehrdad Hosseinzadeh, Omit Chanda, and Yang Wang. Crowd counting using scale-aware attention networks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1280–1288. IEEE, 2019.

[26] Sultan Daud Khan and Saleh Basalamah. Sparse to dense scale prediction for crowd couting in high density crowds. *Arabian Journal for Science and Engineering*, 46(4):3051–3065, 2021.

[27] V. Ranjan, H. Le, and M. Hoai. Iterative crowd counting. In *ECCV*, 2018.

[28] X. Cao, Z. Wang, Y. Zhao, and F. Su. Scale aggregation network for accurate and efficient crowd counting. In *ECCV*, pages 734–750, 2018.

[29] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5099–5108, 2019.

[30] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Learning scales from points: A scale-aware probabilistic model for crowd counting. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 220–228, 2020.

[31] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7661–7669, 2018.

[32] Zhi-Qi Cheng, Jun-Xiu Li, Qi Dai, Xiao Wu, and Alexander G Hauptmann. Learning spatial awareness to improve crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6152–6161, 2019.

[33] Muming Zhao, Jian Zhang, Chongyang Zhang, and Wenjun Zhang. Leveraging heterogeneous auxiliary tasks to assist crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12736–12745, 2019.

[34] Jia Wan and Antoni Chan. Adaptive density map generation for crowd counting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1130–1139, 2019.

[35] Junyu Gao, Qi Wang, and Xuelong Li. Pcc net: Perspective crowd counting via spatial convolutional network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3486–3498, 2019.

[36] Vishwanath A Sindagi and Vishal M Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 1–6. IEEE, 2017.

[37] Jiang Liu, Chenqiang Gao, Deyu Meng, and Alexander G Hauptmann. Decidenet: Counting varying density crowds through attention guided detection and density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5197–5206, 2018.

[38] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6142–6151, 2019.

[39] B. Wu and R. Nevatia. Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. *International Journal of Computer Vision*, 75(2):247–266, 2007.

[40] M. Patzold, R. H. Michael, and T. Sikora. Counting people in crowded environments by fusion of shape and motion information. In *AVSS*, pages 157–164, 2010.

[41] L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht. Learning to count with regression forest and structured labels. In *ICPR*, pages 2685–2688, 2012.

[42] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. In *CVPR*, 2017.

[43] E. Walach and L. Wolf. Learning to count with cnn boosting. In *ECCV*, pages 660–676, 2016.

[44] X. Shen, Y.-C. Chen, X. Tao, and J. Jia. Convolutional neural pyramid for image processing. *arXiv preprint arXiv:1704.02071*, 2017.

[45] C. Arteta, V. Lempitsky, J. A. Noble, and A. Zisserman. Interactive object counting. In *ECCV*, pages 504–518, 2014.

[46] Yaocong Hu, Huan Chang, Fudong Nian, Yan Wang, and Teng Li. Dense crowd counting from still images with convolutional neural networks. *Journal of Visual Communication and Image Representation*, 38:530–539, 2016.

[47] Elad Walach and Lior Wolf. Learning to count with cnn boosting. In *ECCV*, 2016.

[48] Cong Zhang, Hongsheng Li, Xiaogang Wang, and Xiaokang Yang. Cross-scene crowd counting via deep convolutional neural networks. In *CVPR*, 2015.

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

[50] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. Deep people counting in extremely dense crowds. In *ACM Multimedia*, 2015.

[51] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, 2016.

[52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[53] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4(2), 2012.

[54] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. In *CVPR*, 2016.

[55] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[56] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[57] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Aistats*, volume 9, pages 249–256, 2010.

[58] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

[59] Senjian An, Wanquan Liu, and Svetha Venkatesh. Face recognition using kernel ridge regression. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7. IEEE, 2007.

[60] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[61] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241, 2015.

[62] Ming Liu, Yang Liu, Jue Jiang, Zhenwei Guo, and Zenan Wang. Crowd counting with fully convolutional neural network. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 953–957. IEEE, 2018.

[63] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[64] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7268–7277, 2018.

[65] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[66] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi. Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 552–568, 2018.

[67] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[68] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1451–1460. IEEE, 2018.

[69] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

[70] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2547–2554, 2013.

[71] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[72] Jing Shao, Kai Kang, Chen Change Loy, and Xiaogang Wang. Deeply learned attributes for crowded scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4657–4666, 2015.

[73] Deepak Babu Sam, Neeraj N Sajjan, R Venkatesh Babu, and Mukundhan Srinivasan. Divide and grow: Capturing huge diversity in crowd images with incrementally growing cnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3618–3626, 2018.

[74] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5382–5390, 2018.

[75] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting,

density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–546, 2018.

[76] Zhikang Zou, Xinxing Su, Xiaoye Qu, and Pan Zhou. Da-net: Learning the fine-grained density distribution with deformation aggregation network. *IEEE Access*, 6:60745–60756, 2018.

[77] Jun Sang, Weiqun Wu, Hongling Luo, Hong Xiang, Qian Zhang, Haibo Hu, and Xiaofeng Xia. Improved crowd counting method based on scale-adaptive convolutional neural network. *IEEE Access*, 7:24411–24419, 2019.

초 록

본 학위논문에서는 군중의 혼잡도와 사람의 크기를 고려한 딥러닝 기반의 새로운 군중 밀도 추정 방법을 제시합니다. 군중 밀도 추정은 지능형 감시 시스템의 중요한 과제들 중 하나입니다. 군중 밀도 추정을 사용하여 공공 보안 및 안전에 대한 관심 영역을 쉽게 표시할 수 있습니다. 또한 이를 이용하면 보행자 감지, 추적 등 연산 부담이 높은 고급 컴퓨터 비전 알고리즘이 지능형 감시 시스템에 효과적으로 적용하는 것을 도울 수 있습니다.

군중 밀도 추정에 딥 러닝이 도입된 후 대부분의 연구는 훈련 이미지로 군중 밀도 맵을 추정하는 네트워크를 학습하기 위해 컨볼루션 신경망을 사용하는 관습적인 방식을 따릅니다. 딥 러닝 기반 군중 밀도 추정 연구는 네트워크 구조 관점과 훈련 전략 관점의 두 가지 관점으로 나뉠 수 있습니다. 일반적으로 네트워크 구조 관점의 연구에서는 군중을 잘 표현하기 위한 특징을 추출하기 위한 새로운 네트워크 구조를 제안합니다. 반면 훈련 전략 관점에서는 계수 성능을 향상시키기 위해 새로운 훈련 방법론이나 손실 함수를 제안합니다.

본 학위논문에서는 딥러닝 기반 군중밀도 추정에서 두 가지 관점에서 여러 연구를 제안합니다. 특히, 각 사람의 군중 혼잡도와 규모에 따라 풍부한 군중 표현 특성을 갖도록 제안하는 모델을 설계합니다. 선택적 앙상블 네트워크와 계단식 잔여 확장 네트워크의 두 가지 새로운 네트워크 구조를 제안합니다. 또한 군중 밀도 추정을 위한 새로운 손실 함수인 혼잡 인식 베이지안 손실을 제안합니다.

먼저, 정확한 군중밀도 추정과 인원 계수를 위한 선택적 앙상블 딥 네트워크 구조를 제안합니다. 기존 딥 네트워크 기반 방법과 달리 제안된 방법은 지역 밀도 추정을 위해 두 개의 하위 네트워크를 통합합니다. 하나는 희소 밀도 영역 학습용이고

다른 하나는 밀집 밀도 영역 학습용입니다. 두 개의 하위 네트워크에서 지역적으로 추정된 밀도맵은 초기 군중밀도로 추정되며 게이팅 네트워크를 사용하여 앙상블 방식으로 선택적으로 결합됩니다. 초기 밀도맵은 이미지의 컨텍스트 정보를 기반으로 하는 또 다른 하위 네트워크를 사용하여 고해상도 맵으로 개선됩니다. 네트워크 훈련에서 새로운 적응형 손실 체계를 적용하여 혼잡한 지역의 모호성을 해결합니다. 제안된 기법은 밀집도 및 훈련 정도에 따라 밀도 손실과 계수 손실 사이의 가중치를 조정하여 밀도맵 정확도와 계수 정확도를 모두 향상시킵니다.

두 번째로, 스케일이 다른 다중 확장 컨볼루션 블록으로 구성된 새로운 군중밀도 추정 네트워크 구조를 제안합니다. 제안된 네트워크 구조는 소규모 확장 컨볼루션은 각 사람의 중심 영역 밀도를 정확히 추정하는 반면 대규모 확장 컨볼루션은 사람의 주변 영역 밀도를 잘 추정한다는 경험적 분석에서 비롯되었습니다. 군중에 있는 각 사람의 중심에서 주변으로 점차적으로 군중밀도맵을 추정하기 위해 여러 확장된 컨볼루션 블록이 작은 확장 컨볼루션 블록에서 큰 블록으로 계단식으로 훈련됩니다.

마지막으로, 사람 규모와 군중 희소성을 고려한 새로운 혼잡 인식 베이지안 손실 방법을 제안합니다. 딥 러닝 기반 군중 밀도 추정은 군중 계산의 정확도를 크게 향상시킬 수 있습니다. 베이지안 손실 방법은 손으로 만든 지상 진실 밀도와 잡음이 있는 주석의 필요성이라는 두 가지 문제를 해결하지만 혼잡한 장면에서 정확하게 계산하는 것은 여전히 어려운 문제입니다. 군중 장면에서 사람의 외모는 각 사람의 크기('사람 크기')에 따라 바뀝니다. 또한 국부 영역의 희소성('군중 희소성')이 낮을수록 군중 밀도를 추정하기가 더 어렵습니다. 장면 기하정보를 기반으로 '사람 크기'를 추정한 다음 추정된 '사람 크기'를 사용하여 '군중 희소성'을 추정합니다. 추정된 '사람 크기' 및 '군중 희소성'은 새로운 혼잡 인식 베이지안 손실 방법에서 사용되어 점 주석의 교사 표현을 개선합니다.

제안된 밀도 추정기의 효율성은 널리 사용되는 군중 계산 벤치마크 데이터 세트에 대한 최첨단 방법과의 비교 실험을 통해 검증되었습니다. 제안된 방법은 다양한 감시 환경에서 최첨단 밀도 추정기보다 우수한 성능을 달성했습니다. 또한 제안된 모든 군중 밀도 추정 방법에 대해 여러 자가비교 실험을 통해 각 구성 요소의 효율성을 검증했습니다.

주요어: 군중 밀도 감지, 군중 계수, 장면이해, 영상감시
학번: 2014-21714