ZiZoNet: A Zoom-In and Zoom-Out Mechanism for Crowd Counting in Static Images

Usman Sajid

Submitted to the graduate degree program in Department of Electrical Engineering and Computer Science and the Graduate Faculty of the University of Kansas in partial fulfillment of the requirements for the degree of Master of Science.

Dr. Guanghui Wang, Chairperson

Committee members

Dr. Bo Luo

Dr. Heechul Yun

Date defended: May 13, 2019

The Thesis Committee for Usman Sajid certifies
that this is the approved version of the following thesis :

ZiZoNet: A Zoom-In and Zoom-Out Mechanism for Crowd Counting in Static Images

Dr. Guanghui Wang, Chairperson

Date approved: May 13, 2019

Abstract

As people gather during different social, political or musical events, automated crowd analysis can lead to effective and better management of such events to prevent any unwanted scene as well as avoid political manipulation of crowd numbers. Crowd counting remains an integral part of crowd analysis and also an active research area in the field of computer vision. Existing methods fail to perform where crowd density is either too high or too low in an image, thus resulting in either over-estimation or underestimation. These methods also mix crowd-like cluttered background regions (e.g. tree leaves or small and continuous patterns) in images with actual crowd, resulting in further crowd overestimation. In this work, we present a novel deep convolutional neural network (CNN) based framework ZiZoNet for automated crowd counting in static images in very low to very high crowd density scenarios to address above issues. ZiZoNet consists of three modules namely Crowd Density Classifier (CDC), Decision Module (DM) and Count Regressor Module (CRM). The test image, divided into 224x224 patches, passes through the CDC module that classifies each patch to a class label (no-crowd, low-crowd, medium-crowd, high-crowd). Based on the CDC information and using either heuristic Rule-set Engine (RSE) or machine learning based Random Forest based Decision Block (RFDB), DM decides which mode (zoom-in, normal or zoom-out) this image should use for crowd counting. CRM then performs patch-wise crowd estimate for this image accordingly as decided or instructed by the DM module. Extensive experiments on three diverse and challenging crowd counting benchmarks (UCF-QNRF, ShanghaiTech, AHU-Crowd) show that our method outperforms current state-of-the-art models under most of the evaluation criteria.

Acknowledgements

I would like to thank my thesis advisor, Dr. Guanghui Wang, for his support and guidance throughout this research work. I would also like to thank my thesis committee members, Dr. Bo Luo and Dr. Heechul Yun, for their valuable time and suggestions.

I would also like to thank my family without whom support this would not have been possible.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we define the problem and challenges, motivation and background, and our contributions. In the first section, background and motivation have been explained in detail, followed by briefly providing overview of used approach and explaining contributions of this work. Lastly, we go through the organization of this work. Most part of this chapter has been taken from the introduction section of our work [32].

1.1 Motivation

Crowd counting remains an integral part of crowd analysis. While masses converge to huge gatherings like Hajj, sporting and musical events or political rallies, automated crowd count can lead to better and effective management of such events and prevent any unwanted incident [16]. Crowd counting is an active research problem due to different challenges pertaining to large perspective, huge variance in scale and image resolution, severe occlusions and dense crowd-like cluttered background regions. Manual crowd counting subjects to very slow and inaccurate results due to the complex issues as mentioned above.

1.2 Background and Challenges

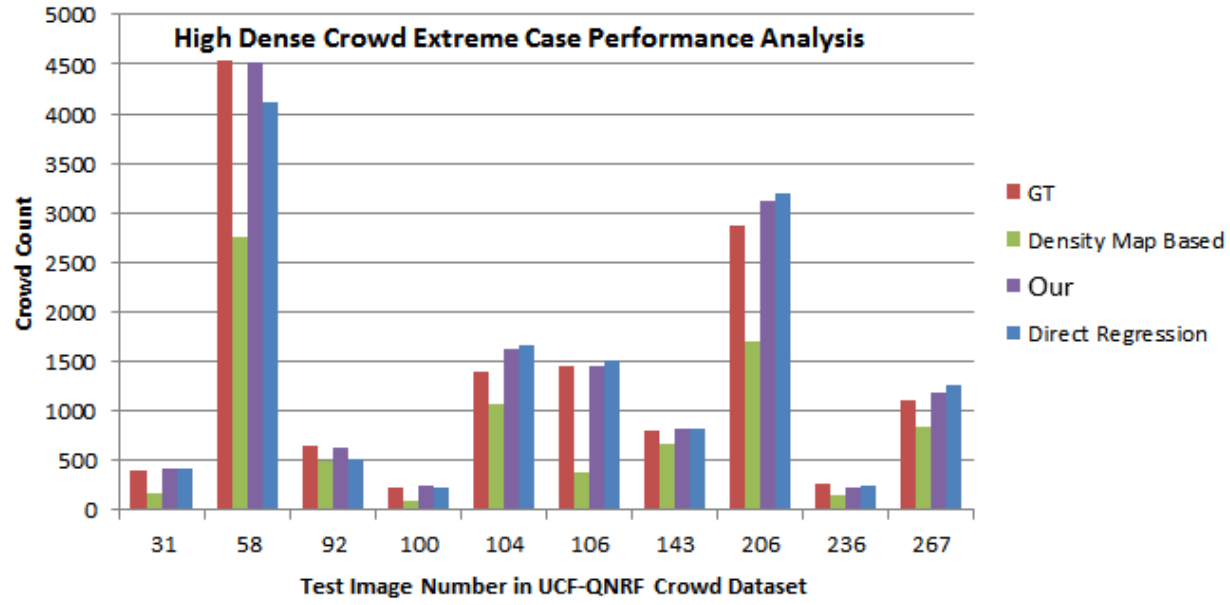
To obtain accurate, fast and automated crowd counting results, CNN-based approaches have been proposed that achieve superior performance over traditional approaches [8, 10, 40]. CNN-based methods can be broadly classified into three categories; regression-based, detection-based, and density map estimation methods. Regression-based methods [38] directly regress the count from

(a) GT=0, Regr.=75 Ours=0, Density=72 | (b) GT=0, Regr.=18 Ours=0, Density=19 | (c) GT=4535, Regr.=4109 Ours=4523, Density=2759 | (d) GT=704, Regr.=861 Ours=708, Density=1017

Figure 1.1: Direct regression and Density map [3] methods overestimate in case of crowd-like cluttered background patches as in (a) and (b), where there is no crowd at all. Similarly, these methods highly underestimate or overestimate in two extreme cases, where most crowd patches belong to either high or low-crowd count as in (c) and (d) respectively, as compared to the ground truth (GT).

the input image. However, these CNN regressors alone cannot handle huge diversity in the crowd images varying from very low to very high. CNN detection-based methods [13, 30] first detect persons in the image and then sum all detection results to yield the final crowd count estimate. Detection-based methods perform well in low crowd images but could not be generalized well to high-density crowd images as detection fails miserably in such cases due to very few pixels per head or person. Density map estimation methods [36, 20, 37] generate density map values, with one value for each image pixel. The final estimate is then calculated by summing all density map values. These methods do not rely on localizing crowd but rather on estimating crowd density in each region of the crowd image. Density map estimation methods outperform other approaches and current state-of-the-art methods mostly belong to this category. However, density per pixel estimation remains a huge challenge as indicated in [29] due to large variations in the crowd density across different images. This naturally leads to a question: *In which scenarios these methods may fail and why?*

One key issue with regression and density map methods is that they only rely on direct count estimate and density map estimation per pixel for the input image respectively, thus, they may get subjected to large crowd count for cluttered background image patches. As shown in Figure 1.1, models [20] based on these methods consider this 224×224 image patch as a crowd patch and



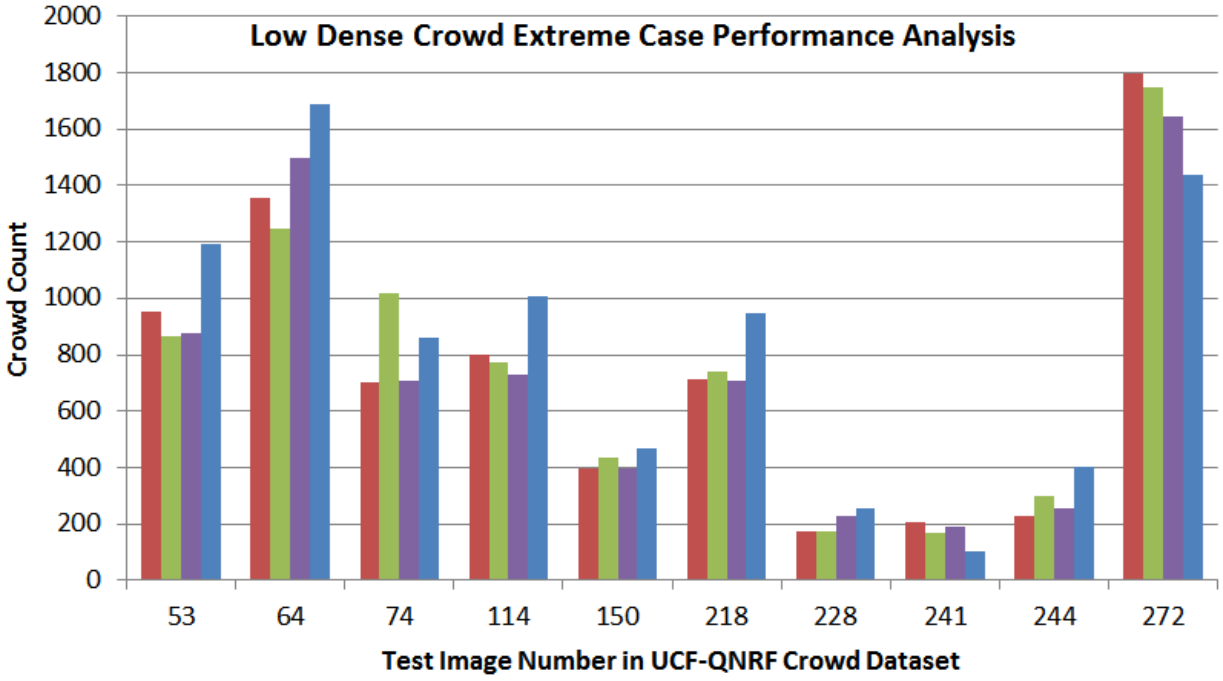


Figure 1.2: Upper and lower graphs compare ten cases each, belonging to high dense and low dense extreme respectively for Density map [3], DenseNet [18] based direct regression and our method. As shown, other models either highly underestimate or overestimate, whereas the proposed method remains the closest to the ground truth (GT) bar in most cases.

make false estimates, making the system unreliable as similar patterns are bound to occur in many practical scenarios.

In addition, we observe that both types of methods perform well for images which contain most crowd patches with neither low nor high crowd density. Problem arises when images have most crowd patches with either high or low-density crowd. Due to the limitation in handling such practical diversity in crowd density, these methods may either highly underestimate or overestimate the crowd count in these two extreme cases, as shown in Figure 1.1. To further explain this phenomenon, we analyze ten such cases for both extremes separately from very recent UCF-QNRF dataset [20] on the state-of-the-art density map method [3, 20] and direct regression-based method as shown in Figure 1.2. It can be observed that, in both extreme cases, the crowd estimates are either highly overestimated or underestimated due to the limitations as discussed above. So, it remains a challenge to get a stable crowd count not only in normal crowd density cases, but also in the case of very low or very high crowd density images.

Another challenge or limitation with crowd counting research is smaller datasets with images ranging from few hundreds to one or two thousand in number. As CNN-based approaches rely on huge training datasets for their training, so it puts a constraint on proposed methods to-date. Keeping that in mind, mostly state-of-the-art existing methods uses image patches instead of actual images to approach crowd counting problem in general. So, main challenges we focus on during this work, are as follows:

- Large variation across crowd density across different images that results in overestimation or underestimation.

- Crowd-like cluttered background regions in images that result in further overestimation.

- Availability of few training images, usually only few hundred for training purpose.

1.3 Overview of the Work

To solve the above mentioned fundamental issues in crowd counting, we propose a modular approach that comprises of a Crowd Density Classifier (CDC), a novel Decision Module (DM), and a Count Regressor Module (CRM). The input image is first sub-divided into fixed-size patches (224×224) and fed to the CDC module that contains a deep CNN classifier to perform a four-way classification (low, medium, high-density, and no-crowd) on each patch. The classification module eliminates any crowd-like background patches (no-crowd) from the test image and feeds the information about the number of patches belonging to each of the no-crowd, low, medium and high-density classes to the Decision Module (DM) using an accumulator. DM uses either the machine learning based RFDB module or heuristic-based RSE module to determine if the image belongs to a case of low, normal or high density. Based on the DM decision, this image is then divided into fixed-size patches using one of three independent image patch-making modules $(Z_{in}, Normal, Z_{out})$. The image, belonging to low-density extreme case, gets divided into patches using the zoom-out (Z_{out}) patch-maker, high-density extreme case via zoom-in (Z_{in}) patch-making block, and images of normal case split into patches using normal ($Normal$) patch-maker. These patches are then routed one by one to the patch-wise count regressor ($COUNTER$) for crowd estimate and the image total crowd count is obtained by summing all patches count.

The Z_{in} block divides each input patch into four 112×112 patches, and then up-scales each patch by $2\times$ before routing each patch to the count regressor. Intuitively, this module is further zooming-in into the image and looking in-detail all patches by using $1/2$ input patch size instead of the original 224×224 patches. Similarly, zoom-out patch-maker divides the input image into 448×448 patches, and down-scales each patch by $2\times$ as it is dealing with the image containing low-density crowd patches mostly. The normal case image directly employs the original 224×224 patch size with no up-scaling or down-scaling.

1.4 Contributions of Thesis

The main contributions of this work include:

- This work reveals and analyzes the fact that extremely high and low dense crowd images greatly influence the performance of the state-of-the-art regression and density map based methods for crowd counting.

- A novel strategy is proposed to address the problem of counting in highly varying crowd density images by first classifying the images into either one of the extreme cases (of very low or very high density) or a normal case, and then feeding the images to specifically designed patch-makers and crowd regressor for counting.

- A novel rule-set engine is developed to determine whether the image belongs to an extreme case. For images of extremely high density, a zoom-in strategy is developed to look into more details of the image; while for images of low-density extreme, a zoom-out based regression is employed to avoid overestimate.

- We created three new datasets, each from the corresponding crowd counting benchmark, for the training and testing of different machine learning algorithms to classify an image as normal, high or low-dense extreme case using its patches classification count. These manually verified datasets will facilitate the researchers in analyzing complex crowd diversity, which is at the core of the crowd analysis.

- ZiZoNet works without using any density maps. Consequently, it eliminates the limitation of density map estimation per pixel problem.

The proposed ZiZoNet scheme is thoroughly evaluated on three benchmarks: UCF-QNRF [20], ShanghaiTech [45], and AHU-Crowd [17]. The experimental results demonstrate the effectiveness

and generality of the proposed strategy and rule-sets, which are never realized for crowd counting. The overall performance of the proposed model outperforms the state-of-the-art approaches on most of the evaluation criteria.

1.5 Organization

Rest of the thesis is organized as follows. Sec. 2 introduces the related work in crowd counting field, followed by a detailed description of the proposed framework in Sec. 3. Sec. 4 elaborates the implementation details with focus on training and tuning discussion. Extensive experiments, including ablation studies and analysis, are presented in Sec. 4.2. In the end, the thesis is concluded in Sec. 5.

Chapter 2

Related Work

Broadly, crowd counting approaches can be classified into pre-CNN (classical) and post-CNN (modern) era techniques as shown in Figure 2.1. Nowadays, most solutions use convolutional neural network (CNN) based network to solve different challenges for crowd counting problem. Most part of this chapter has been taken from the related work section of our work [32].

2.1 Classical Techniques

Crowd counting remains an active research area in computer vision with different challenges related to large perspective, occlusion, cluttered background regions and high variance in crowd density across different images. Earlier work [39, 40, 4, 41, 44] focused on the head or full-body detection for counting using handcrafted features for detectors learning. These methods failed in case of high dense images, where it is hard to find such handcrafted features. The approaches were shifted towards regression based counting [8, 10, 31, 9], where a mapping function was learned to directly regress count from local patches of an image. These methods improved the counting process, however, they could not handle huge crowd diversity and also lack awareness about crowd density across all parts of the image.

2.2 CNN-based Techniques

Recently, CNN-based approaches have been widely used [20, 24, 38, 25]. They are broadly categorized into three classes; Counting by detection, counting by direct regression, and counting using density map estimation.

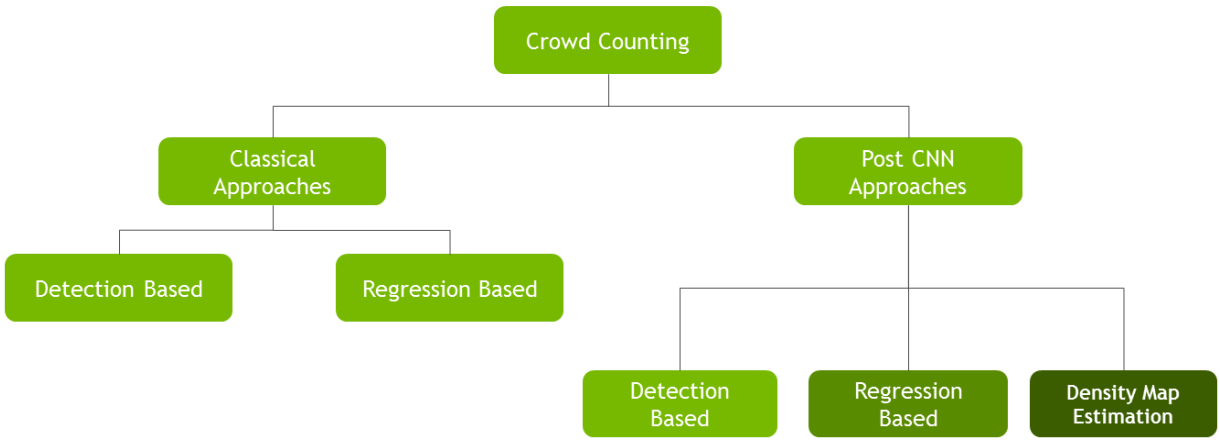


Figure 2.1: Crowd Counting methods category-wise. Most methods nowadays belong to either regression based or density map estimation based CNN methods.

2.2.1 Counting by Detection

CNN-based object detectors [13, 30, 26] detect each person in the image, and the final count is then calculated by summing all detections. Detection process usually consists of either head or full body detection for each person. Detection based methods work quite well for low density crowd images where it's easier to detect such objects. But these methods [34, 23] deteriorate in high density and severe occlusion cases, where each head or body only occupies a few pixels.

2.2.2 Counting by Regression

Counting by direct regression methods [38] directly regress the count by learning feature maps from the input image patch. Wang *et al.* [38] proposed an end-to-end AlexNet [22] based regressor for crowd count. These methods alone cannot handle huge diversity in different crowd images and have to be incorporated with some specialized guiding or controlling mechanism. Our work is based on this key idea that regression based method used with some specialized controlling scheme enables huge crowd diversity handling across different images.

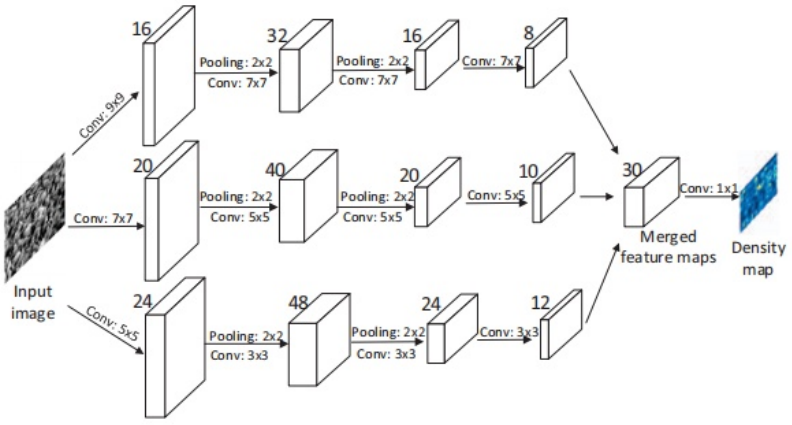


Figure 2.2: Multi-column convolutional neural network (MCNN) based on density map estimation method [45].

2.2.3 Density Map Estimation based Crowd Counting

Density map estimation methods [20, 24, 7, 36, 25, 45] learn to map crowd density per pixel of an image without localizing the counts. The final estimate is calculated by summing all density estimations. An example of such scheme is shown in Figure 2.2 as proposed by Zhang *et al.* [45], consisting of a three-column CNN architecture (MCNN) to handle crowd diversity across images. Each column is designed to handle different scales using different receptive field sizes. Sindagi *et al.* [37] extended the idea of MCNN to incorporate contextual information for high-quality density maps generation. Recently, Sam. *et al.* [33] proposed SwitchCNN which routes each input patch to one of three independent CNN regressors using a switch CNN classifier. Based on the classification and regression idea, Sindagi *et al.* [36] designed a Cascaded-MTL that estimates count for the whole image by using cascaded 10-way classification prior and final density map estimation.

2.3 Image-wise vs Patch-wise Techniques

Generally, crowd counting methods estimate crowd count using either whole image directly or each patch one by one. Crowd counting models, based on whole image estimation and training

from the scratch [21], are subjected to over-fitting due to limited dataset availability (only a few hundred training images). Thus, patch-based models are widely used nowadays. The final sum is computed by adding up all patch count estimates. Liu *et al.* [25] proposed a hybrid approach by incorporating both regression and detection blocks using an attention-guided mechanism to handle low and high-density cases simultaneously. Li *et al.* [24] designed a CSRNet to get multi-scale contextual information by incorporating dilation-based convolutional layers. Idress *et al.* [20] proposed a composition loss based model for simultaneous crowd counting and localization.

2.3.1 Comments on Existing Methods

Existing methods perform worse in extreme cases where most crowd patches belong to either high density or low density. Moreover, these methods lack the ability to fully discard any cluttered background regions in the image, thus resulting in overestimate.

Chapter 3

ZiZoNet Framework

The proposed framework is shown in Figure 3.2, which is composed of three modules namely Crowd Density Classifier (CDC), Decision Module (DM) and Count Regressor Module (CRM). The input image is first sub-divided into 224×224 size patches and each patch then passes through the CDC module for 4-way classification (low, medium, high-density or no-crowd). The accumulator gathers and feeds patch count per class information to the Decision Module. Based on accumulator information and utilizing either Random Forest based Decision Block (RFDB) or heuristic-based Rule-Set Engine (RSE), DM routes this image to one of three specialized patch-making blocks $(Z_{in}, Normal, Z_{out})$ of CRM where the input image is divided into corresponding patches, followed by the crowd estimate for each patch via crowd regressor $(COUNTER)$. Finally, the image crowd count is calculated by summing all patches count. Below we will discuss the details of each module, as well as the rules defined for the two possible extremes. Most part of this chapter has been taken from the proposed approach section of our work [32].

3.1 Crowd Density Classifier (CDC) Module

3.1.1 Description

The CDC module is composed of a deep CNN 4-way classifier that specializes in making a distinction between no-crowd (NC), low-density (LC), medium-density crowd (MC), and high-density crowd (HC) for each input patch. Let X be a test image sub-divided into N patches $[x_1, x_2, ...x_N]$, each with a size of 224×224. The accumulator gathers each patch classification result for the

input image X as follows:

$$P_y \mathrel{+}= 1, \;\; if \; class(x_i) \; = \; y \tag{3.1}$$

for $i = 1, 2, ...N$ and y belongs to either NC, LC, MC or HC class label. In the end, the accumulator passes the patch count per class (PCC_X) of this image to the decision module (DM) as:

$$PCC_X \; = \; \{P_{NC}, P_{LC}, P_{MC}, P_{HC}\} \tag{3.2}$$

where P_{NC}, P_{LC}, P_{MC} and P_{HC} denote the total number of patches being classified as NC, LC, MC and HC respectively of the image X. Patches being classified as NC are discarded, and thus remaining $\{N - P_{NC}\}$ crowd patches are going to be used for final crowd estimate. As a result, the crowd-like cluttered background regions (such as the tree leaves shown in Figure 1.1), which may result in overestimation otherwise, will be eliminated.

3.1.2 Definitions of NC, LC, MC and HC class labels

During experiments for each crowd counting benchmark dataset, we randomly extract patches from its training images for the CDC classifier training and assign a ground truth class label (NC,LC,MC,HC) to each extracted patch. Since these datasets also contain the localization of people, so we generate the ground truth class label for each patch using this information and the maximum people count possible in any image patch of the corresponding dataset. LC class label is assigned to a patch if the ground truth people count for that patch is less than or equal to 5% of the maximum possible count but greater than zero as zero crowd means NC class patch. Similarly, patches with ground truth people count between 5% to 20% of the maximum possible count are assigned the MC class label, while patches containing more than 20% of the maximum people count are labeled as HC category patches. In the end, a total of $90,000$ patches, with an equal amount per class label, are generated for the CDC classifier training in each benchmark setting. Example patches for each class label are shown in Figure 3.3.

Table 3.1: Description of two Rule-Sets: the lower density extreme (Rules 1-4) and the higher density extreme (Rules 5-8). Third column indicates images that are affected the most (in terms of resolution) by that rule. Some rules have much higher tendency to be applied on the Lower Resolution (LR) or Higher Resolution (HR) images, whereas some rules have impact on all types of images (indicated by 'Mix').

Extreme Case Type	**Rule**	**Most Affected Images**	**Description**
Low	1	LR	Image contains *LC* and *NC* patches only.
	2	Mix	Image should have LC patches and no HC patch.
	3	HR	Image has more than 50% patches being classified as *LC* category.
	4	Mix	At most 5% patches belong to *HC* category with at least one patch from *NC* category.
High	5	Mix	Image with all patches belonging to *HC* category only.
	6	Mix	All patches are *MC* category only.
	7	LR	More than 50% patches of the image are from *HC* category.
	8	Mix	Image should have *NC* patches and at least 33% or more from both P_{HC} and P_{MC} category each. Intuitively, first condition of R8 emphasizes the fact that more no-crowd patches shift image towards high dense case, if supported by other given conditions.

3.1.3 CDC Classifier Details

We use DenseNet-201 [18] as our 4-way classifier, as shown in Figure 3.4. It has four dense blocks with transition layers (convolution and pooling) in between them to adjust feature maps size accordingly. The DenseNet-201 has consecutive 1×1 and 3×3 convolutional layers in each dense block in $\{6, 12, 48, 32\}$ sets respectively. At the end of the last dense block, a classification layer is composed of 7×7 global average pooling, followed by $1000 - D$ fully connected layer and the final 4-way softmax classification with cross-entropy loss.

3.2 Decision Module (DM)

The decision module, based on the CDC module output PCC_X, decides if the test image should be treated as a normal image or a low or a high-density extreme case image. DM makes this decision by utilizing one of the two separate and independent decision-making blocks, namely Rule-Set Engine (RSE) and Random Forest based Decision Block (RFDB). RSE is a novel heuristic-based

Algorithm 1 Rule-Set Engine Algorithm for the RSE Module

Input: PCC_X(Patch Count per Class for Test Image X)= {$\mathbf{P_{NC}}$, $\mathbf{P_{LC}}$, $\mathbf{P_{MC}}$, $\mathbf{P_{HC}}$}
Output: Normal or $\mathbf{Z_{in}}$ or $\mathbf{Z_{out}}$
Let $P_{all} = P_{NC} + P_{LC} + P_{MC} + P_{HC}$
if *input patch count satisfies any of following rules* **then** $Output = Z_{out}$
  Rule 1: $if\ P_{HC} + P_{MC} == 0$
  Rule 2: $if\ P_{HC} == 0\ and\ P_{LC} > 0$
  Rule 3: $if\ P_{LC} > (P_{all} * 0.50)$
  Rule 4: $if\ P_{NC} > 0\ and\ P_{HC} <= (P_{all} * 0.05)$
end
else if *input patch count satisfies any of following rules* **then** $Output = Z_{in}$
  Rule 5: $if\ P_{LC} + P_{MC} == 0$
  Rule 6: $if\ P_{LC} + P_{HC} == 0$
  Rule 7: $if\ P_{HC} > (P_{all} * 0.50)$
  Rule 8: $if\ P_{NC} > 0\ and\ P_{MC} >= (P_{all} * 0.33)\ and$
  $P_{HC} >= (P_{all} * 0.33)$
end
else $Output = Normal$

approach which employs the rule-sets to detect if the test image is either an extreme or a normal case, while RFDB is an automated decision-making block based on Random Forest algorithm that learns to map the test image features ($P_{NC}, P_{LC}, P_{MC}, P_{HC}$) to the respective class label (Z_{in}, $Normal$, Z_{out}). We also create new RFDB training datasets, each from corresponding crowd counting benchmark, for the training of RFDB module as explained in Sec. 3.2.3.

3.2.1 Rule-Set Engine (RSE)

The accumulated patch count per class (PCC_X) from CDC module is tested against two different rule-sets to determine if an input image is a case of low or a high density extreme or a normal one so that it can be divided into patches using the most suitable patch-making block $(Z_{in}, Normal, Z_{out})$. The overall goal of RSE is to encourage an image with more number of high-density patches to pass through zoom-in patch-making block (Z_{in}), whereas the image with more number of low-density patches goes through a zoom-out patch-making block (Z_{out}). If the image does not belong to any of the two extreme cases, it will be treated as a normal case that uses the normal patch-maker $(Normal)$.

3.2.1.1 RSE Engine Rules

The RSE module consists of two generalized rule-sets, aiming to detect the images belonging to any of the two extreme cases: the low-density extreme (Rules 1-4) and the high-density extreme (Rules 5-8). As illustrated in Algorithm 1, if no rule applies to the test image X, it will use *Normal* patch-maker, whereas the image satisfying any rule from $(1-4)$ or $(5-8)$ will generate its patches using Z_{out} or Z_{in} patch-making blocks respectively. Each rule is explained in detail in Table 3.1. This table also shows the most affected images by a specific rule in terms of resolution. For example, Rule 7 is highly applicable on relatively lower resolution (LR) images, whereas Rule 2 can affect images of any resolution equally. It is important to note that these rule-sets are used consistently and evaluated across all three publicly available datasets in the experiments, thus demonstrating the generality and efficacy of such rule-sets. In addition, the current rule sets are extendable by adding more rules to refine the classification/decision process. Please note that all parameters in Table 3.1 are chosen empirically.

3.2.2 Random Forest based Decision Block (RFDB)

The scalable rule-sets based decision process yields promising results as demonstrated throughout the experiments in Sec. 4.2. Nevertheless, there are many heuristics to handle and it requires manual input and special attention while inducting new rules. To address this issue, we propose an automated machine learning based approach that learns the decision process by mapping the four features $(P_{NC(\%)}, P_{LC(\%)}, P_{MC(\%)}, P_{HC(\%)})$ to respective class label (Z_{in}, *Normal or* Z_{out}) for each image, where the features denote percentages instead of total image patches belonging to NC, LC, MC and HC classes respectively and labels represent zoom-in, normal and zoom-out patch-making blocks required to generate the patches from the particular input image before proceeding to the count regressor. We employ percentages for features because of the huge variance in resolution across different images in a dataset, which directly influences the features and hence training quality. In addition, since there is no such dataset available for the crowd counting problem to-date that can help in learning this mapping, thus we generate a new RFDB training dataset from

each corresponding benchmark as explained in detail in next subsection. To automate the decision block process, we explored different machine learning classification models and found the random forest-based model to be the most effective as demonstrated in the experiments in Sec. 4.2. Thus, we choose the random forest algorithm and hence named this module as Random Forest based Decision Block.

Random Forest (RF), being a bootstrap aggregation or bagging based ensemble method, can be used both for classification and regression. We employ the RF algorithm to classify the four features $(P_{NC(\%)}, P_{LC(\%)}, P_{MC(\%)}, P_{HC(\%)})$ to a class label of (Z_{in}, Normal or Z_{out}) by building, training and tuning a large collection of de-correlated binary decision trees. Each tree then casts a vote for class prediction for the test sample. Finally, the class label with a majority vote is assigned to that test sample i.e., the input image.

Each RF decision tree t_k is built using a bootstrap sample $BS(t_k)$ which is generated from the training data. Such bootstrap sample is given as:

$$BS(t_k) = \begin{bmatrix} NC_1 & LC_1 & MC_1 & HC_1 & C_1 \\ NC_2 & LC_2 & MC_2 & HC_2 & C_2 \\ NC_3 & LC_3 & MC_3 & HC_3 & C_3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ NC_M & LC_M & MC_M & HC_M & C_M \end{bmatrix} \tag{3.3}$$

for $K = 0, 1, 2, ...N-1$, where N denotes the total number of RF trees. Each row represents one training sample for the tree t_k with the class label as the last entry. We use $N = 100$, which is set empirically as no significant improvement has been observed in performance beyond this number. The trees are grown using the classification and regression (CART) algorithm, where the nodes get split until all leaves become unmixed or contain less than m_{min} samples [1]. We use $m_{min} = 2$ throughout our experiments, thus splitting nodes until they contain either only one sample or become pure. To quantify the quality of a tree node split, *Gini Impurity* has been used

as:

$$Gini\ Impurity_n = \sum_{i=1}^{L=3} -F_i(1-F_i) \tag{3.4}$$

where L denotes the total unique class labels and F_i denotes the frequency of class label i at node n. During testing, each RF tree gives its class prediction for test image X. Final class label is obtained by the majority vote criterion [14] as follows:

$$C_{RF}(X) = majority\ vote\{C_k(X)\}_1^N \tag{3.5}$$

where $C_k(X)$ represents the class prediction by the k^{th} RF tree.

3.2.2.1 Feature Importance Analysis

Feature Importance (FI) depicts the role of each feature in determining the node split and eventually the quality of the RF decision trees building. Features with much lesser FI value can be easily discarded as they do not play any significant role in decreasing the node impurity. As shown in the graph in Figure 3.5, all four features have approximately the same FI values in each RFDB dataset. Thus, we keep and use all four available features $(P_{NC(\%)}, P_{LC(\%)}, P_{MC(\%)}, P_{HC(\%)})$ in all three newly generated RFDB datasets.

3.2.3 Dataset generation for RFDB

The RFDB module learns to map the image extracted features $(P_{NC(\%)}, P_{LC(\%)}, P_{MC(\%)}, P_{HC(\%)})$ to the respective class label (Z_{in}, *Normal or* Z_{out}) using training dataset with the corresponding mapping. No such dataset has been created to-date. Thus, for each benchmark (ShanghaiTech [45], UCF-QNRF [20], AHU [17]), we created a new respective RFDB dataset which contains this mapping.

To create the new RFDB dataset, each training image's required features $(P_{NC(\%)}, P_{LC(\%)}, P_{MC(\%)}, P_{HC(\%)})$ are extracted using ground truth crowd localization information and definitions of class labels (NC, LC, MC, HC) as stated in 3.1, followed by manual verification and ground truth (GT)

class label assignment. To ensure the quality of the generated dataset, each sample entry was then double checked for any inconsistency, duplicates, missing and erroneous cases. For the extracted features, we use percentages instead of the actual number of patches $(P_{NC}, P_{LC}, P_{MC}, P_{HC})$ belonging to each category because of the huge resolution difference across the images within each benchmark dataset.

3.2.3.1 RFDB Training Datasets Statistics

For each of the three crowd counting benchmarks, we create the corresponding RFDB dataset using its respective training images. For instance, in the case of ShanghaiTech dataset (300 training images), we generate the new 300 samples RFDB dataset with each entry being created using one of the respective training image, followed by manual verification that also includes removal or modification of inconsistent entries. In total, 220 and 812 samples are finalized for two RFDB datasets based on ShanghaiTech [45] and UCF-QNRF [20] benchmarks respectively. For AHU [17] based RFDB dataset, 90 out of 96 available entries are kept on average with 5-fold cross-validation. The graph in Figure 3.6 shows the percentage of each class label in all three newly created RFDB datasets.

3.3 Count Regressor Module (CRM)

The CRM module comprises of three independent patch-making blocks and a deep CNN count regressor $(COUNTER)$. The decision module routes the test image to one of these patch-makers for dividing it into 224×224 patches after required up-scaling or down-scaling, followed by the crowd count for each image patch via the count regressor $(COUNTER)$. The regressor employs DenseNet-201 [18] inspired architecture with a single neuron after the fully connected layer to directly regress the crowd count. Mean squared error (MSE), as defined below, has been employed

as the loss function for the count regressor c :

$$L_c = \frac{1}{N}\sum_{i=1}^{N}(F(X_i,\Theta) - Y_i)^2 \tag{3.6}$$

where N is the number of training patches per batch, Y_i is the ground truth crowd count for the input patch X_i, and F is the function that maps the input patch X_i to the crowd count with learnable parameters Θ.

3.3.1 Zoom-in based Patch Maker (Z_{in})

Ideally, the decision module (DM) routes the image, with most crowd patches being classified as high-density crowd, to this patch-maker. The image, using this patch-maker, is further sub-divided into equal 112×112 patches, and then up-scaled by $2\times$ before proceeding to the count regressor for each patch crowd count. Intuitively, it looks into each patch in detail by estimating the count on smaller zoomed-in highly crowded patches. In this way, it greatly stabilizes and improves the count estimate for high-density images, where other methods may either underestimate or overestimate too much due to fixed patch sizes, as demonstrated in the experiments Sec. 4.2.

3.3.2 Zoom-out based Patch Maker (Z_{out})

This block is responsible for handling the low-density extreme case images as detected and routed by the decision module. Z_{out} takes 448×448 original patches of the test image X, down-scales them by 2 times, and feeds each resultant patch to the CDC classifier to eliminate any no-crowd patches, as shown in Figure 3.7. The count estimate for each crowd patch is then computed through CRM count regressor $(COUNTER)$ followed by the image total count estimate, which is the sum of all patches crowd counts. In other words, it assists the count regressor by using larger area per input patch (448×448 down-scaled to 224×224) which alleviates the overestimation problem.

3.3.3 Normal case

In *Normal* case, the images are divided into 224×224 size patches with no up- or down-scaling before patch-wise count regression. It is also worth mentioning that there is no need to explicitly look for and eliminate any no-crowd patches in case of *Normal* and Z_{in} case images as such background patches are automatically removed during the CDC module classification process, and thus we can also reuse the remaining CDC module crowd patches in both these cases for crowd estimate.

3.4 Comments on the Proposed Architecture

So, using the modular architecture (Figure 3.2) as discussed in detail, we aimed to solve the key challenges with crowd counting research area. In short, we used a hybrid approach where we decide about on mode of counting using the whole image (image-wise) but perform the actual image count by estimating crowd on each of its patch (patch-wise). Usually, existing methods rely only on either image-wise or patch-wise approach. Furthermore, we focused on stabilizing the crowd counting in cases where images are very highly crowded or very low crowded. Other methods result in overestimation and underestimation in such scenarios. We also addressed and discarded any crowd-like cluttered and complex regions in images that may result in further crowd overestimation.

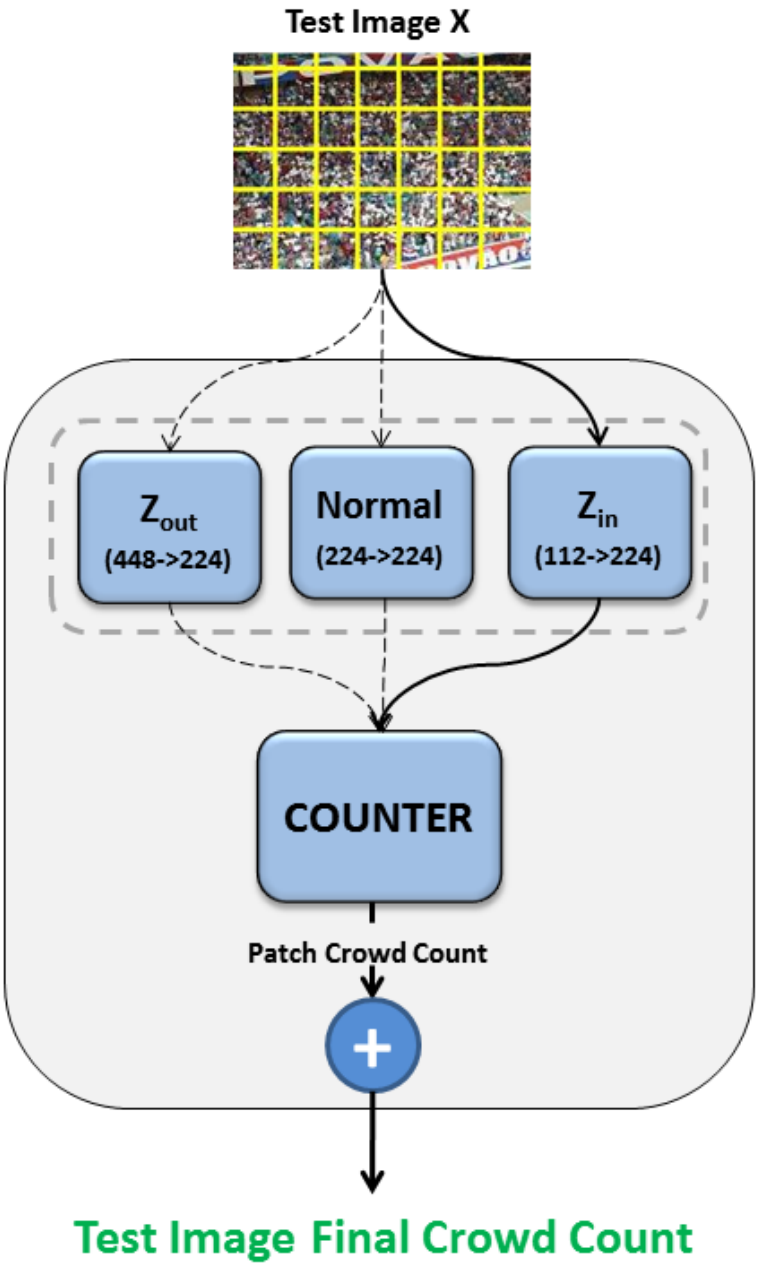


Figure 3.1: ZiZoNet Crowd Counting Base.

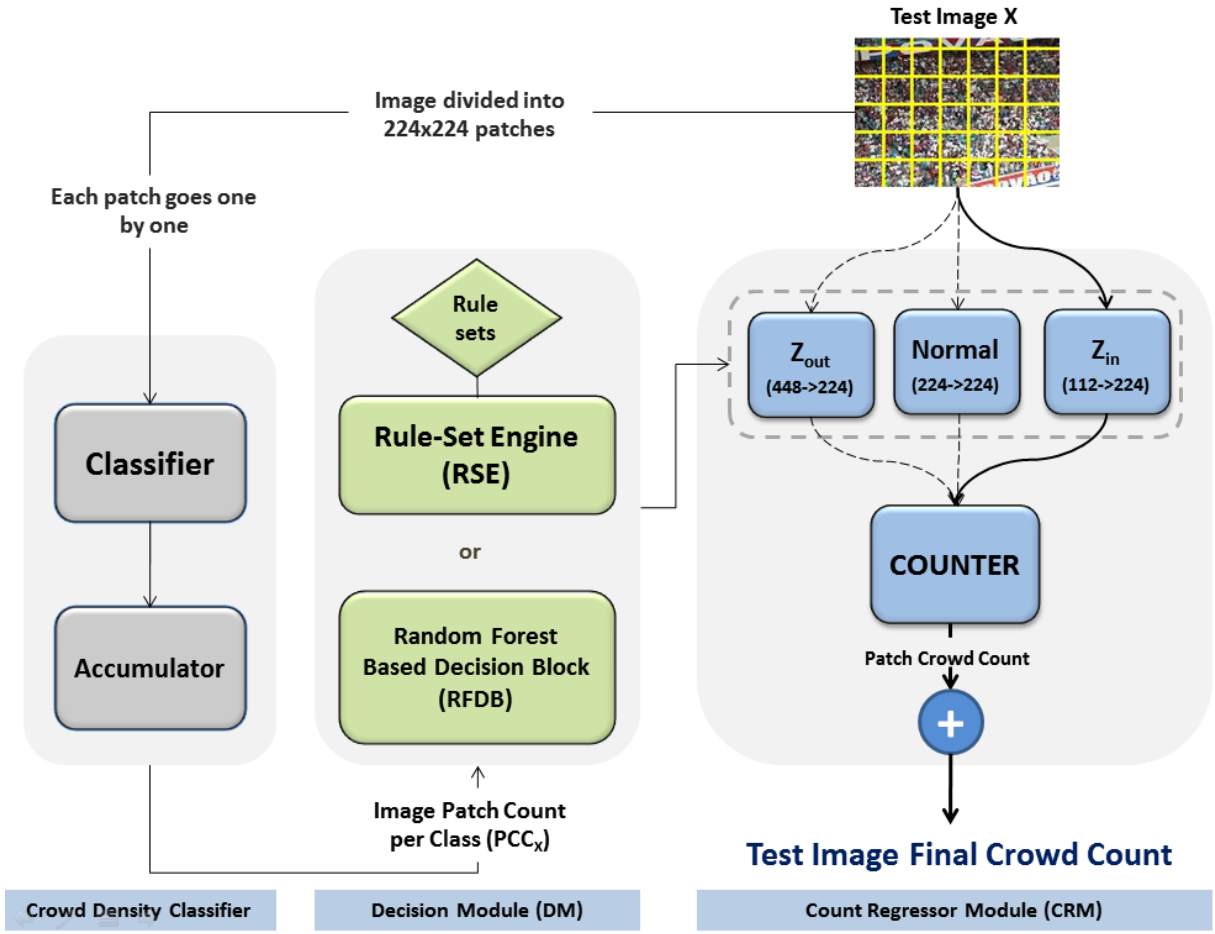


Figure 3.2: ZiZoNet architecture. The test image X, divided into 224×224 patches, first passes through the Crowd Density Classifier (CDC) module, which discards no-crowd patches as classified by the robust 4-way DenseNet classifier. The accumulator stores patch count per class (PCC_X) of this image. Decision Module (DM), based on CDC module output and using either autonomous RFDB or heuristic-based RSE module, decides whether this image should be divided into all normal (*Normal*) patches or make all either zoom-in (Z_{in}) or zoom-out (Z_{out}) based patches before proceeding to the patch-based regressor ($COUNTER$) for each patch crowd count. Image final crowd estimate is then obtained by summing all patches count.

no-crowd(NC) low-crowd(LC) medium-crowd(MC) high-crowd(HC)

Figure 3.3: Actual patches being used for the CDC classifier training. They belong to one of the four class labels (NC, LC, MC, HC) based on the definition.

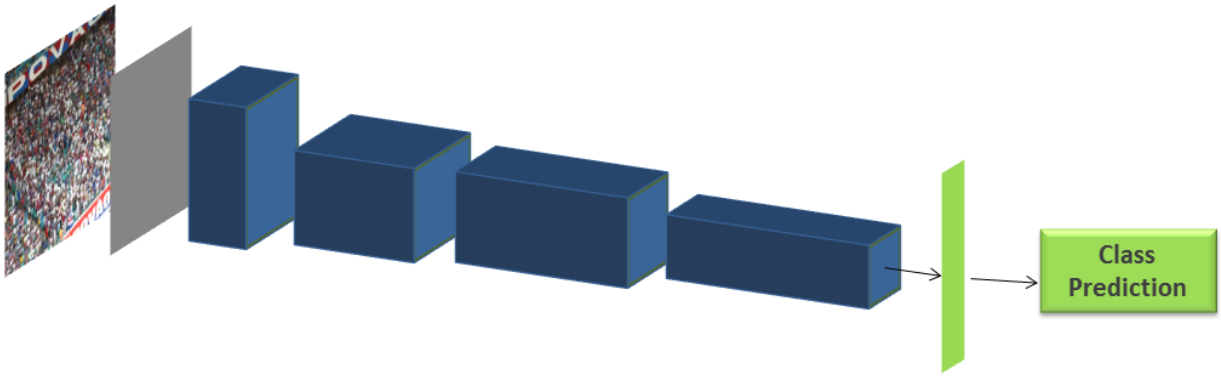


Figure 3.4: Densenet-201 architecture used for 4-way crowd density classification. Blue blocks represent Dense Blocks 1, 2, 3 and 4 from left to right, followed by a fully connected layer and final softmax 4-way classification. 224×224 size input patch is expected.

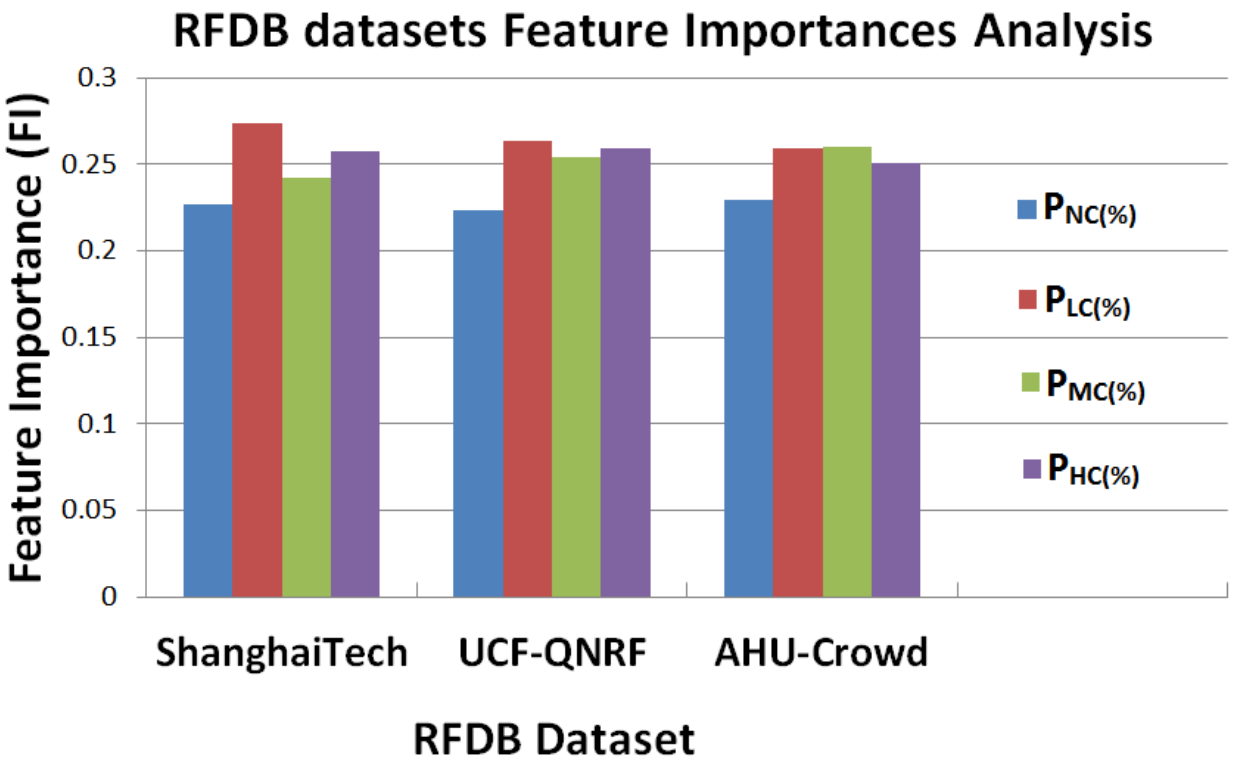


Figure 3.5: Graph shows the Feature Importance (FI) analysis of features from new RFDB datasets created using the corresponding benchmarks (ShanghaiTech, UCF-QNRF and AHU-Crowd). As shown by the FI value, each of the four features plays an important and equal role in maintaining its respective RFDB dataset quality.

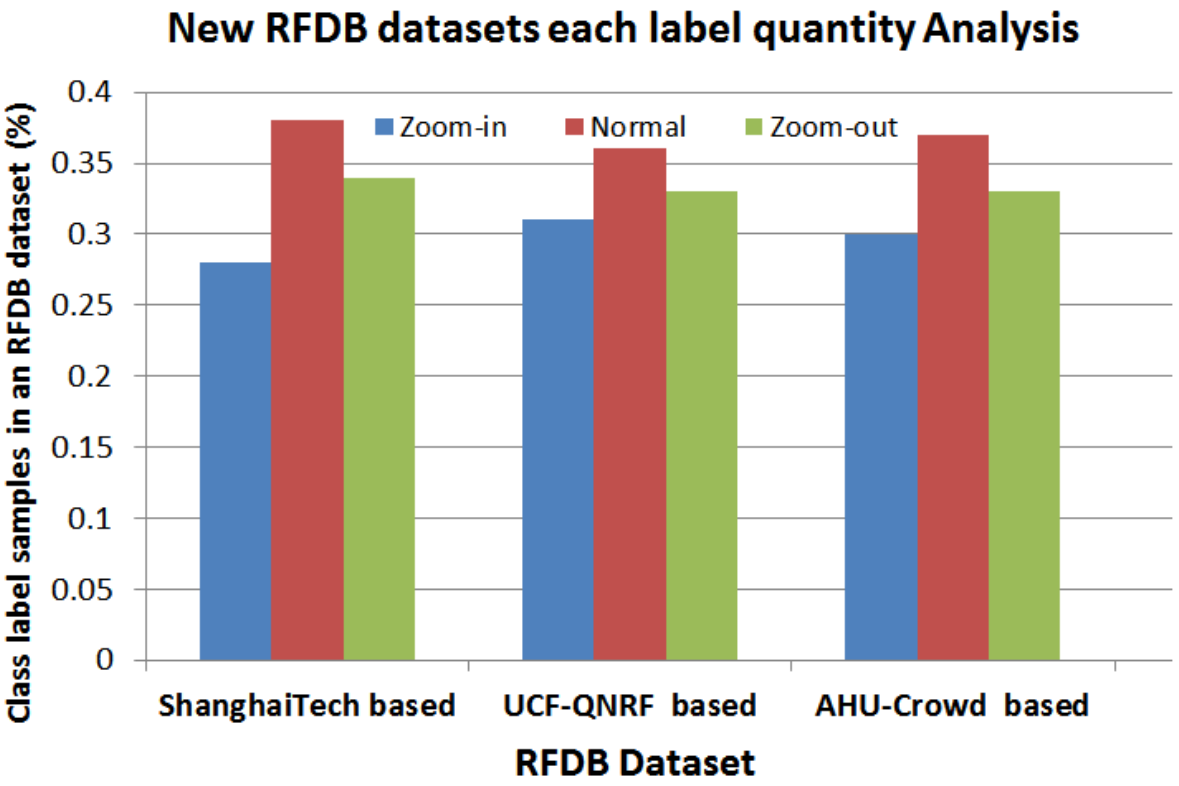


Figure 3.6: Graph depicts the total samples per class $(\%)$ in the new RFDB datasets, each created from corresponding benchmark dataset as indicated by the horizontal axis.

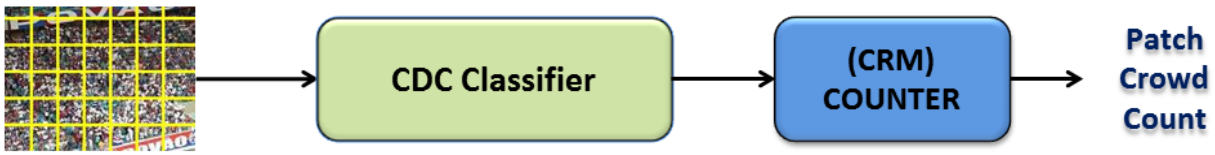


Figure 3.7: Workflow in case if the patch maker Z_{out} is selected by the decision module for the test image count estimate. The input image is divided into 448×448 patches, then down-scaled by 2 times and fed to the CDC classifier to eliminate no-crowd patches. Crowd patches are then routed to the $COUNTER$ for each patch crowd estimate.

Chapter 4

Experiments and Results

In this chapter, we provide the technical implementation and evaluation details followed by experimental results on three benchmark datasets. Most part of this chapter has been taken from the implementation details and experiments sections of our work [32].

4.1 Implementation Details

4.1.1 Training Details

The CDC classifier and the count regressor ($COUNTER$) expect fixed size patch of 224×224 as the input. For both modules, we randomly extract 112×112, 224×224 and 448×448 patches from the training images. Around 90,000 such patches with mixed crowd numbers are generated for each of these modules. The count regressor is trained for 80 epochs with Adam optimizer and a batch size of 16 and starting learning rate of 0.001, decreased by half after every 20 epochs. The classifier employs the stochastic gradient descent (SGD) based optimization with multi-step learning rate starting at 0.1 and decreased by half after 25% and 50% epochs with 80 epochs in total. For each dataset, around 10% training data has been used for validation as recommended in the corresponding literature. For the random forest algorithm in RFDB, we utilize machine learning library scikit-learn for python programming. The Random Forest model was trained using 100 RF decision trees, where each RF tree is trained using the bootstrapped sample with *Gini Imprity* as node split quality criterion. 10% of the training data has been used for validation in case of each RFDB dataset.

Table 4.1: Benchmark datasets (used in the experiments) statistics.

Dataset	Images	Annotations	Min	Max.	Avg.
UCF-QNRF [20]	1535	1,251,642	65	12865	815
ShanghaiTech Part-A [45]	482	241,677	33	3139	501
AHU-Crowd [17]	107	45,807	58	2201	428

Table 4.2: Comparison of ZiZoNet with the state-of-the-art methods on the UCF-QNRF [20] dataset. Methods with '*' do not use density maps at all. Both versions of our method outperform the state-of-the-art on most of the evaluation criteria.

	MAE	MNAE	RMSE
Idrees et al. [19]*	315	0.63	508
MCNN [45]	277	0.55	426
Encoder-Decoder [6]	270	0.56	478
CMTL [36]	252	0.54	514
SwitchCNN [33]	228	0.44	445
Resnet101 [15]*	190	0.50	277
Densenet201[18]*	163	0.40	226
CL [20]	132	0.26	**191**
ZiZoNet-RSE*	130	0.23	204
ZiZoNet-RFDB*	**128**	**0.20**	201

4.1.2 Evaluation Details

In order to make a fair and consistent comparison with other methods, we employ three evaluation metrics namely Mean Absolute Error (*MAE*), Mean Normalized Absolute Error (*MNAE*) and Root Mean Squared Error (*RMSE*) defined as below:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|Y_i - \hat{Y}_i| \tag{4.1}$$

$$MNAE = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y_i - \hat{Y}_i|}{Y_i} \tag{4.2}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y}_i)^2} \tag{4.3}$$

Table 4.3: Ablation experiments on UCF-QNRF [20] dataset emphasize importance of zoom-in, zoom-out patch-making blocks and associated rules in ZiZoNet-RSE. The first eight rows depict the effect of removing one rule at a time on MAE, MNAE and RMSE while next three rows demonstrate the effect without using the zoom-in (Z_{in}), zoom-out (Z_{out}) and both zoom-in and zoom-out blocks respectively, followed by the method with original setting in last row. I_{Zin}, I_N, I_{Zout} indicate the total images handled by zoom-in, normal and zoom-out patch-makers respectively before proceeding to the count regressor.

Without	MAE	MNAE	RMSE	I_{Zin}	I_N	I_{Zout}
R1	130.6	0.23	204	75	101	158
R2	130	0.23	204	75	115	144
R3	138.4	0.27	230	75	106	153
R4	130	0.23	204	75	127	132
R5	130	0.23	204	75	97	162
R6	130	0.23	204	75	97	162
R7	130	0.23	204	75	97	162
R8	140.7	0.23	219	8	164	162
Z_{in}	141.3	0.23	220	0	172	162
Z_{out}	150.0	0.31	244	75	259	0
Z_{in} &Z_{out}	160.1	0.30	250	0	334	0
-	**130**	**0.23**	**204**	75	97	162

where N denotes the total number of test images, and Y_i *and* $\hat{Y}_i$ are the ground truth and the estimated counts respectively for the test image i.

4.2 Experiments

In this section, we demonstrate both quantitative and qualitative results from extensive experiments on three benchmark datasets: UCF-QNRF [20], ShanghaiTech [45], and AHU-Crowd [17]. These datasets contain images with huge crowd variance, different camera perspective and complex cluttered background regions. Details about each benchmark are given in Table 4.1.

Two different versions of the proposed model, the Rule-Set Engine (**ZiZoNet-RSE**) based and the automated RFDB module (**ZiZoNet-RFDB**) based version, are being compared separately with the state-of-the-art techniques throughout this section. Both ZiZoNet versions give almost identical and much better performance under most of the evaluation criteria on the three benchmark datasets.

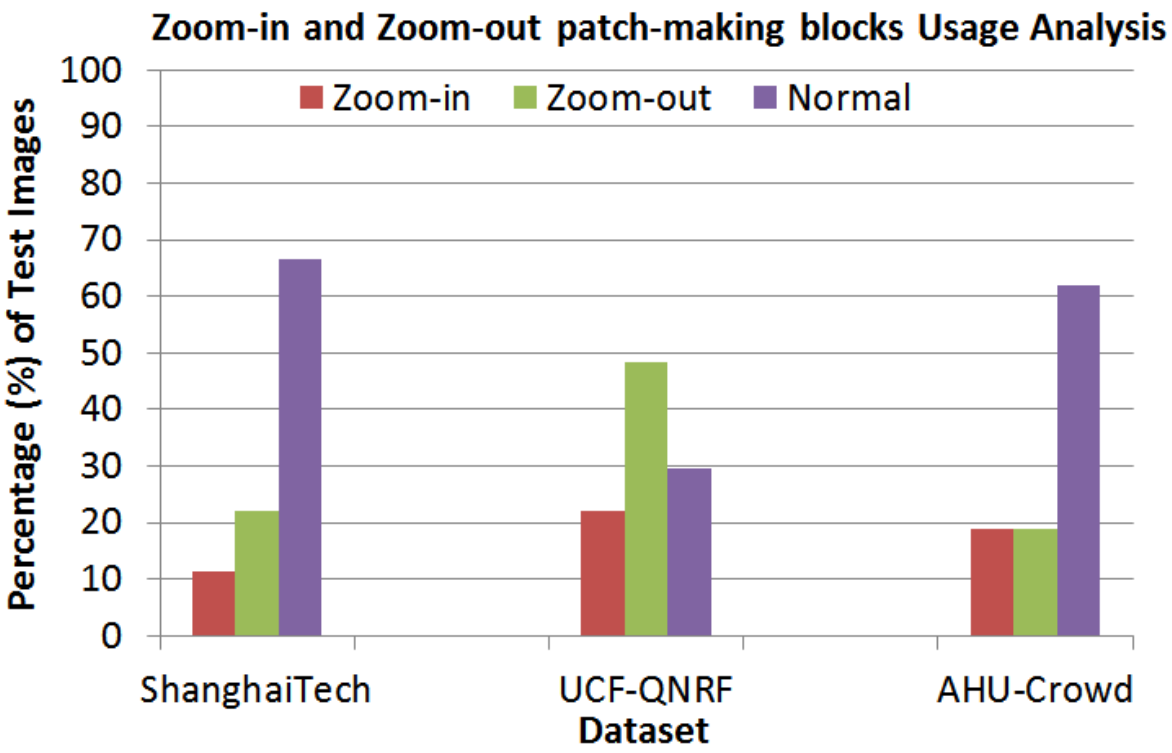


Figure 4.1: Quantitative importance of zoom-in and zoom-out blocks and given rule sets. For each benchmark, at least 11% and as high as 48.5% test images pass through one of these specialized patch-makers before patch-wise count regression, demonstrating the value and effectiveness of such blocks and associated rules.

4.2.1 Experiments on UCF-QNRF Dataset

The dataset was recently published by Idrees *et al.* [20], which is a challenging and the first dataset of its kind. On one hand, it contains images with resolution as high as (6666×9999) and as low as (300×377); on the other hand, crowd count per image ranges from a maximum value of $12,865$ to a minimum count of 65. The total number of annotations in this dataset is $1,251,642$, indicating the level of crowd complexity. It contains 1535 images in total, out of which 1201 and 334 images are used for training and testing respectively. We compare ZiZoNet with the state-of-the-art methods and tabulate the results in Table 4.2. It is evident that both versions of our method outperform all other approaches in terms of MAE and MNAE; while performing competitively closer to the best in terms of RMSE.

Table 4.4: Comparison of ZiZoNet with the state-of-the-art approaches on the ShanghaiTech [45] dataset, where '*' indicates methods not using density maps at all. Our method performs the best on every evaluation criteria.

	MAE	MNAE	RMSE
Zhang et al. [43]	181.8	-	277.7
MCNN [45]	110.2	-	173.2
Cascaded-MTL [36]	101.3	0.279	152.4
Switch-CNN [33]	90.4	-	135.0
CP-CNN [37]	73.6	-	106.4
CSRNet [24]	68.2	-	115.0
IG-CNN [5]	72.5	-	118.2
L2R [27]	72.0	-	106.6
ICC [29]	68.5	-	116.2
SA-Net [7]	67.0	-	104.5
Deep-NCL [35]	73.5	-	112.3
Densenet201[18]*	79.3	0.224	118.9
ZiZoNet-RSE*	66.6	0.197	**94.5**
ZiZoNet-RFDB*	**66.0**	**0.190**	97.5

In order to evaluate the influence of different rules, we perform the ablation experiments, as shown in Table 4.3. We analyze the effect of all rules (R1 to R8) by removing them one at a time in ZiZoNet-RSE version of the proposed method. As shown in the results, doing so greatly decreases the performance of our method, thus demonstrating the importance of those rules. We also analyze the effect of removing both or either of the zoom-in and zoom-out patch-makers in the experiments. From the results in Table 4.3, it is evident that both modules play an effective role in improving the overall performance of our method. The last three columns of Table 4.3 show the number of the test images passed through the zoom-in, normal and zoom-out patch-makers respectively. In the original setting, 75 ($\sim 22\%$) images passed through the zoom-in patch-maker, whereas the zoom-out block handled 162 ($\sim 48.5\%$) images and normal patch-maker was used only for 97 ($\sim 29.5\%$) images, showcasing quantitative importance of these extreme case handlers, as shown in Figure 4.1. We also compare the crowd estimate of ten test images each, for both extreme cases with DenseNet[18] direct regression and the state-of-the-art CL [20] density map method. Our method performs much better in both cases, as shown in Figure 1.2.

Table 4.5: Ablation experiments on the ShanghaiTech [45] dataset show quantitative importance of the zoom-in, zoom-out patch-making blocks and associated rules in ZiZoNet-RSE. The first eight rows depict the effect of removing one rule at a time. Next three rows demonstrate the results without using the zoom-in (Z_{in}), zoom-out (Z_{out}) and both zoom-in and zoom-out blocks respectively, followed by the method with original setting in the last row. I_{Zin}, I_N, I_{Zout} indicate the total images handled by zoom-in, normal and zoom-out blocks respectively before proceeding to the counter.

Without	MAE	MNAE	RMSE	I_{Zin}	I_N	I_{Zout}
R1	66.8	0.199	94.8	21	135	26
R2	66.8	0.198	94.6	21	122	39
R3	67.2	0.198	94.7	21	124	37
R4	66.6	0.197	94.5	21	121	40
R5	69.4	0.200	103.7	15	127	40
R6	69.1	0.210	101.8	19	123	40
R7	66.8	0.200	97.2	20	122	40
R8	66.8	0.197	94.6	15	127	40
Z_{in}	74.9	0.200	116.4	0	142	40
Z_{out}	69.1	0.210	96.5	21	161	0
Z_{in} & Z_{out}	78.3	0.210	118.9	0	182	0
-	**66.6**	**0.197**	**94.5**	21	121	40

4.2.2 Experiments on ShanghaiTech Dataset

The ShanghaiTech part A dataset contains a total of 482 images with 241,677 annotations, randomly collected from the internet, with a split of 300 and 182 images for training and testing respectively. We compare our method with the state-of-the-art methods as shown in Table 4.4. The results show that our method outperforms all other methods on every evaluation metric with significant improvement from 0.224 to 0.190($\sim 15\%$) in terms of MNAE and from 104.5 to 94.5($\sim 9.6\%$) in case of RMSE.

The proposed rules (R1-R8) play an important and effective role in the performance improvement of ZiZoNet-RSE version of our method as shown in Table 4.5, where we remove each rule one at a time. It is clear that the error increases by removing these rules. In the same table, We also analyze the effect of removing the zoom-in and zoom-out blocks separately and together. As expected, the performance plunges dramatically as error increases without using them. The last three columns show the number of test images passing through the zoom-in, normal and zoom-out patch-makers respectively. In the original setting, 21 ($\sim 11.5\%$), 121 ($\sim 66.5\%$) and 40 ($\sim 22\%$)

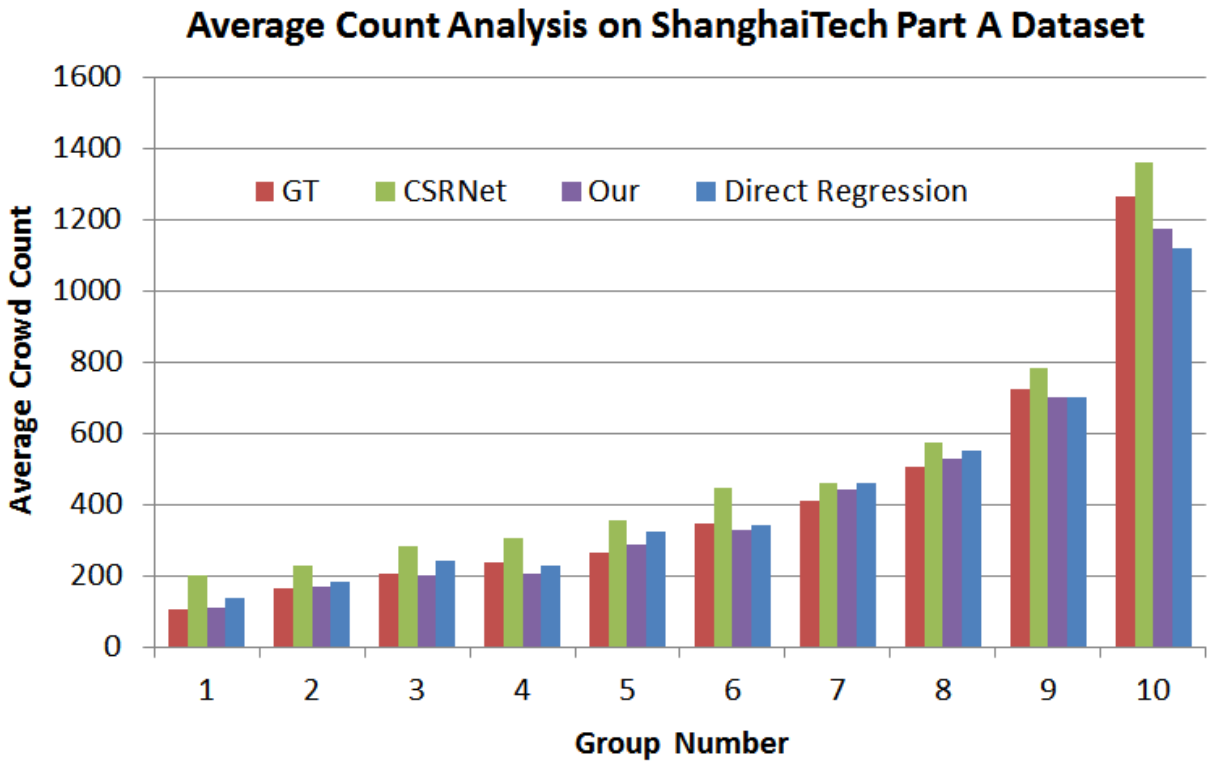


Figure 4.2: 182 test images are divided into ten groups with total crowd count in each group increasing from left to right. Each group contains 18 images except group number 10. Vertical axis indicates average count for each group. It is evident that ZiZoNet remains closer to the ground truth (GT) bar in most cases as compared to the state-of-the-art methods.

images are handled by the zoom-in, normal and zoom-out blocks respectively, thus proving the quantitative importance of all of them and associated rules in ZiZoNet-RSE, as shown in Figure 4.1. In Figure 4.2, we analyze the performance of our method on the average count across image groups with different total crowd counts. As compared with the state-of-the-art methods, ZiZoNet performs the best in most cases.

4.2.3 Experiments on AHU-Crowd Dataset

AHU-Crowd [17] dataset contains 107 images with $45,807$ human annotations. The crowd count ranges from 58 to 2201 per image. As per the standard being followed for this dataset [17], we performed 5-fold cross-validation and evaluated our method using the same three evaluation met-

GT=704, DR=861
Ours=708, Density[3]=1017

GT=1443, DR=1516
Ours=1443, Density[3]=388

GT=4535, DR=4109
Ours=4523, Density[3]=2759

GT=297, DR=474
Ours=299, Density[24]=457

GT=961, DR=997
Ours=996, Density[24]=1022

GT=1366, DR=1425
Ours=1384, Density[24]=1445

Figure 4.3: Some examples of the good qualitative results on the UCF-QNRF [20] and ShanghaiTech [45] datasets. Each result also shows the estimates of DenseNet [18] Direct Regression (DR) and the Density map method as a comparison.

GT=353, DR=493
Ours=489, Density[3]=476

GT=1668, DR=1629
Ours=1476, Density[3]=1717

GT=249, DR=159
Ours=140, Density[24]=174

GT=199, DR=321
Ours=413, Density[24]=413

Figure 4.4: Some examples of the bad qualitative results on the UCF-QNRF [20] and ShanghaiTech [45] datasets. Each result also shows the estimates of DenseNet [18] Direct Regression (DR) and the Density map method as a comparison.

Table 4.6: Comparison of ZiZoNet with the state-of-the-art on the AHU-Crowd [17] dataset, where '*' indicates methods without using density maps. Our method outperforms previous approaches on all evaluation metrics.

	MAE	MNAE	RMSE
Haar Wavelet [28]	409.0	0.912	-
DPM [12]	395.4	0.864	-
BOW–SVM [11]	218.8	0.604	-
Ridge Regression [10]	207.4	0.578	-
Hu et al. [17]	137	0.365	-
DSRM [42]	81	0.199	129
Densenet201[18]*	87.6	0.295	124.9
ZiZoNet-RSE*	79.3	0.198	121
ZiZoNet-RFDB*	**74.9**	**0.190**	**111**

Table 4.7: ZiZoNet performance analysis on ShanghaiTech and UCF-QNRF benchmarks using different ML classification algorithms in the RFDB block of Decision Module (DM). As shown, top five results indicate best performance by the Random Forest algorithm, thus, justifying its usage in the RFDB module.

	ShanghaiTech			UCF-QNRF		
	MAE	MNAE	RMSE	MAE	MNAE	RMSE
Random Forest	**66.0**	**0.190**	**97.5**	**128**	**0.20**	**201**
ExtraTrees	70.7	0.20	102.8	135	0.22	214
GradientBoosting	72.9	0.22	119.0	137	0.22	222
AdaBoost	75.0	0.22	105.31	151	0.24	265
Logistic Regression	78.9	0.23	119.6	177	0.24	279

rics. ZiZoNet outperforms all other methods as shown in Table 4.6. It is worth-mentioning that ZiZoNet decreases MAE and MNAE significantly by $\sim 7.5\%$ (81 to 74.9) and $\sim 4.5\%$ (0.199 to 0.190) respectively, whereas RMSE decreases drastically by $\sim 11.2\%$ (124.9 to 111).

4.2.4 Qualitative Results

In Figures 4.3 and 4.4, we show some good and bad case qualitative results respectively from UCF-QNRF and ShanghaiTech datasets. We also compare our results with the ground truth (GT), DenseNet [18] Regression (DR) and the state-of-the-art density map methods. In each row, the first three cases demonstrate the good results followed by two bad estimates. The bad case results happen mostly due to the test image being detected as wrong extreme case type by the decision

Figure 4.5: Qualitative results of some test images patches being classified correctly as no-crowd (NC), low-crowd (LC), medium-crowd (MC) or high-crowd (HC) by the CDC classifier as shown for each category column-wise.

module (DM). We also show some visual results to demonstrate the qualitative performance of the CDC classifier in Figure 4.5.

4.2.5 RFDB Algorithm Selection

In this paper, we adopt the Random Forest algorithm for the RFDB module. In practice, other machine learning-based classification algorithms can also be employed. In order to choose the best one for our system, we experimented with different classifiers to select the appropriate decision-

making algorithm. The results based on the ShanghaiTech and UCF-QNRF datasets are shown in Table 4.7. We observe that ensemble based methods perform better on our relatively smaller and imbalanced RFDB datasets as they prevent over-fitting and high variance by combining several machine learning techniques. After evaluation, the Random Forest appears to be the best choice as the RFDB algorithm as shown in Table 4.7, where the top five best results justify the selection of the Random Forest algorithm. For these experiments, we used machine learning library scikit-learn for python programming [2].

Chapter 5

Conclusion and Future Work

In this work, we have proposed a novel zoom-in and zoom-out based mechanism for effective and accurate crowd counting in highly diverse images. We propose to employ a decision module to detect the extreme high and low dense cases, where most state-of-the-art regression and density map based methods perform worse. The cluttered background regions are also discarded using the rigorous deep CNN 4-way classifier. Without using any density maps at all, ZiZoNet outperforms the state-of-the-art approaches on three benchmark datasets, thus proving the effectiveness of the proposed model. We also created three new training datasets for the training of different machine learning algorithms to learn the crowd diversity. This will help the researchers working in crowd analysis field to explore the problem further. In this work, we addressed following crowd counting challenges:

- Large variation across crowd density across different images that results in overestimation or underestimation.

- Crowd-like cluttered background regions in images that result in further overestimation.

- Availability of few training Images.

In future, we aim to make the quality of three new RFDB datasets better by exploring additional dimensions and adding more features or density levels. We also look forward to work-with and analyze different revisions of the proposed framework with different networks being used as classifiers and regressors. End-to-end approach is also the most desirable form in deep learning

research, but it seems inevitable in crowd counting problem to use modular approach to handle huge crowd diversity. Most state-of-the-art and end-to-end models fail to address crowd counting challenges alone. But in future, we will try to explore some end-to-end mechanisms for this problem. We also aim to incorporate this approach in some other applicable domains e.g. crowd tracking, crowd anomaly detection etc.

For fair comparison, the source code and the datasets will be available on the Author's website. Most part of this section has been taken from the conclusion section of our work [32].

References

[1] Scikit sklearn library. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed: April 13, 2019].

[2] Scikit sklearn library. https://scikit-learn.org/stable/supervised_learning.html. [Accessed: April 27, 2019].

[3] Web link. http://crcv.ucf.edu/mtayyab/cc/crowd_counting.php.

[4] Agustin, O. C. & Oh, B.-J. (2011). People counting using object detection and grid size estimation. In *International Conference on Future Generation Communication and Networking* (pp. 244–253).: Springer.

[5] Babu Sam, D., Sajjan, N. N., Venkatesh Babu, R., & Srinivasan, M. (2018). 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* (pp. 3618–3626).

[6] Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12), 2481–2495.

[7] Cao, X., Wang, Z., Zhao, Y., & Su, F. (2018). Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 734–750).

[8] Chan, A. B. & Vasconcelos, N. (2009). Bayesian poisson regression for crowd counting. In *IEEE International Conference on Computer Vision* (pp. 545–551).

[9] Chan, A. B. & Vasconcelos, N. (2012). Counting people with low-level features and bayesian regression. *IEEE Transactions on Image Processing*, 21(4), 2160–2177.

[10] Chen, K., Loy, C. C., Gong, S., & Xiang, T. (2012). Feature mining for localised crowd counting. In *BMVC*, volume 1 (pp.$\tilde{3}$).

[11] Csurka, G., Dance, C., Fan, L., Willamowski, J., & Bray, C. (2004). Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1 (pp. 1–2).

[12] Felzenszwalb, P., McAllester, D., & Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, CVPR* (pp. 1–8).

[13] Girshick, R. (2015). Fast R-CNN. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

[14] Hastie, T., Tibshirani, R., & Friedman, J. (2009). The elements of statistical learning: data mining, inference, and prediction, springer series in statistics.

[15] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

[16] Helbing, D. & Mukerji, P. (2012). Crowd disasters as systemic failures: Analysis of the love parade disaster. *EPJ Data Science*, 1(1), 7.

[17] Hu, Y., Chang, H., Nian, F., Wang, Y., & Li, T. (2016). Dense crowd counting from still images with convolutional neural networks. *Journal of Visual Communication and Image Representation*, 38, 530–539.

[18] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *CVPR*, volume 1 (pp.$\tilde{3}$).

[19] Idrees, H., Saleemi, I., Seibert, C., & Shah, M. (2013). Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2547–2554).

[20] Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., & Shah, M. (2018). Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 532–546).

[21] Kang, K. & Wang, X. (2014). Fully convolutional neural networks for crowd segmentation. *arXiv preprint arXiv:1411.4464*.

[22] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097–1105).

[23] Li, W., Li, H., Wu, Q., Meng, F., Xu, L., & Ngan, K. N. (2019). Headnet: An end-to-end adaptive relational network for head detection. *IEEE Transactions on Circuits and Systems for Video Technology*.

[24] Li, Y., Zhang, X., & Chen, D. (2018). CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1091–1100).

[25] Liu, J., Gao, C., Meng, D., & Hauptmann, A. G. (2018a). DecideNet: counting varying density crowds through attention guided detection and density estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5197–5206).

[26] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In *European conference on computer vision* (pp. 21–37).: Springer.

[27] Liu, X., van de Weijer, J., & Bagdanov, A. D. (2018b). Leveraging unlabeled data for crowd counting by learning to rank. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7661–7669).

[28] Oren, M., Papageorgiou, C., Sinha, P., Osuna, E., & Poggio, T. (1997). Pedestrian detection using wavelet templates. In *cvpr*, volume 97 (pp. 193–199).

[29] Ranjan, V., Le, H., & Hoai, M. (2018). Iterative crowd counting. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 270–285).

[30] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).

[31] Ryan, D., Denman, S., Fookes, C., & Sridharan, S. (2009). Crowd counting using multiple local features. In *Digital Image Computing: Techniques and Applications, DICTA* (pp. 81–88).

[32] Sajid, U., Sajid, H., Wang, H., & Wang, G. (2019). Zoomcount: A zooming mechanism for crowd counting in static images. *IEEE Transactions on Circuits and Systems for Video Technology (In Review)*.

[33] Sam, D. B., Surya, S., & Babu, R. V. (2017). Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1 (pp.6̃).

[34] Shami, M., Maqbool, S., Sajid, H., Ayaz, Y., & Cheung, S.-C. S. (2018). People counting in dense crowd images using sparse head detections. *IEEE Transactions on Circuits and Systems for Video Technology*.

[35] Shi, Z., Zhang, L., Liu, Y., Cao, X., Ye, Y., Cheng, M.-M., & Zheng, G. (2018). Crowd counting with deep negative correlation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 5382–5390).

[36] Sindagi, V. A. & Patel, V. M. (2017a). CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)* (pp. 1–6).

[37] Sindagi, V. A. & Patel, V. M. (2017b). Generating high-quality crowd density maps using contextual pyramid CNNs. In *2017 IEEE International Conference on Computer Vision (ICCV)* (pp. 1879–1888).

[38] Wang, C., Zhang, H., Yang, L., Liu, S., & Cao, X. (2015). Deep people counting in extremely dense crowds. In *Proceedings of the 23rd ACM international conference on Multimedia* (pp. 1299–1302).

[39] Wang, M. & Wang, X. (2011). Automatic adaptation of a generic pedestrian detector to a specific traffic scene. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 3401–3408).

[40] Wu, B. & Nevatia, R. (2005). Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *IEEE International Conference on Computer Vision* (pp. 90–97).

[41] Xu, M., Ge, Z., Jiang, X., Cui, G., Zhou, B., Xu, C., et al. (2019). Depth information guided crowd counting for complex crowd scenes. *Pattern Recognition Letters*.

[42] Yao, H., Han, K., Wan, W., & Hou, L. (2017). Deep spatial regression model for image crowd counting. *arXiv preprint arXiv:1710.09757*.

[43] Zhang, C., Li, H., Wang, X., & Yang, X. (2015). Cross-scene crowd counting via deep convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 833–841).

[44] Zhang, Y., Zhou, C., Chang, F., & Kot, A. C. (2018). Attention to head locations for crowd counting. *arXiv preprint arXiv:1806.10287*.

[45] Zhang, Y., Zhou, D., Chen, S., Gao, S., & Ma, Y. (2016). Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 589–597).