Counting in the 2020s: Binned Representations and Inclusive Performance Measures for Deep Crowd Counting Approaches

Thesis submitted in partial fulfillment of the requirements for the degree of

Masters of Science

in

Electronics and Communication Engineering by Research

by

Sravya Vardhani Shivapuja
2019702008
sravya.vardhani@research.iiit.ac.in

International Institute of Information Technology
Hyderabad - 500 032, INDIA
June 2022
CERTIFICATE

It is certified that the work contained in this thesis, titled “Counting in the 2020s: Binned Representations and Inclusive Performance Measures for Deep Crowd Counting Approaches” by Sravya Vardhani Shivapuja, has been carried out under my supervision and is not submitted elsewhere for a degree.

__________________________
Date

Adviser: Prof. Ravi Kiran Sarvadevabhatla
(IIIT Hyderabad)

Co-Adviser: Prof. Ganesh Ramakrishnan
(IIT Bombay)
To My Parents and Friend
Acknowledgments

I would like to take this opportunity to extend my gratitude to Professor Ravi Kiran and Professor Ganesh Ramakrishnan who have guided my work. Their meetings were always insightful and brimming with constructive feedback. They have helped me grow and understand the problem of the topic inside out. Together our exploration in research helped us see the obvious loopholes that the community seemed to miss. Overall the experience was very stimulating and I will treasure these conversations for life. I am exceptionally fortunate to be guided under these excellent professors.

I would also like to extend my thanks to Mansi, Divij, Ayush, Likhitha and Ashwin who were great collaborators for the work. Their dedication and hardwork brought this work to fruition. Additionally, I would like to thank my friends like Swetha and Sreenivas who supported me throughout the entire journey. Without their valuable suggestions I would not have been able to complete this thesis.

More importantly, I would like to thank iHub-Data for recognizing my work and providing fellowship for it. Mainly, Harish from iHub-Data was very helpful and friendly and I extend my thanks to him. I am extremely grateful for receiving this fellowship.
Abstract

Crowd counting is an important task in security, surveillance and monitoring. There are many competitive benchmark datasets available in this domain. The data distribution in the crowd counting datasets show a heavy-tailed and discontinuous nature. This nature of the dataset is majorly ignored while building solutions to this problem. However, the skew in datasets contradicts few assumptions made by the stages of the training pipeline. As a consequence of the skew in the dataset, unacceptably large standard deviation wrt to the customarily used performance measures (MAE, MSE) is observed. To address these issues, this thesis provides modifications that incorporate the dataset skew in training and evaluation pipelines.

In the training pipeline, to enable principled and balanced minibatch sampling, a novel smoothed Bayesian binning approach is presented that stratifies the entire count range. Further, these strata are sampled to construct uniform minibatches. The optimization is upgraded with a novel strata-aware cost function that can be readily incorporated into the existing crowd counting deep networks.

In the evaluation pipeline, as an alternative to the customary evaluation MAE, this thesis provides three alternative evaluation measures. Firstly, a strata-level performance in terms of mean and standard deviation gives range specific insights. Secondly, relative error perspective is brought in by using a novel Thresholded Percentage Error Ratio (TPER). Lastly, a localization included counting error metric Grid Average Mean absolute Error (GAME) is used to evaluate the different networks.

In this thesis, it is shown that proposed binning-based modifications retain their superiority wrt the novel strata-level performance measure. Overall, this thesis contributes a practically useful training pipeline and detail-oriented characterization of performance for crowd counting approaches.
## Contents

| Chapter | Page |
|---------|------|
| 1       | 1    |
| 1.1     | 1    |
| 1.2     | 3    |
| 1.3     | 5    |
| 1.4     | 5    |
| 2       | 6    |
| 2.1     | 6    |
| 2.2     | 8    |
| 2.2.1   | 8    |
| 2.2.2   | 13   |
| 3       | 14   |
| 3.1     | 14   |
| 3.1.1   | 15   |
| 3.1.2   | 15   |
| 3.1.3   | 16   |
| 3.1.4   | 16   |
| 3.2     | 16   |
| 3.3     | 16   |
| 3.3.1   | 16   |
| 3.3.2   | 17   |
| 3.3.3   | 18   |
| 3.3.4   | 19   |
| 3.3.5   | 21   |
| 3.4     | 22   |
| 3.5     | 23   |
| 4       | 27   |
| 4.1     | 27   |
| 4.2     | 27   |
| 4.3     | 28   |
| 4.4     | 30   |

vii
5 Experiment and Analysis

5.1 Results

5.1.1 Strata-level statistics

5.1.1.1 Dataset-wise

5.1.1.2 Network-wise

5.1.2 Aggregate statistics

5.1.3 Percentage Error based results

5.1.3.1 Comparative results over different datasets

5.1.3.2 Robustness of Networks across datasets

5.1.4 Localization Error based results

5.2 Ablations

6 Conclusions and Future Work

6.1 Conclusions

6.2 Future Scope

Bibliography
## List of Figures

| Figure | Page |
|--------|------|
| 1.1 Crowd Counting sample image: Occlusions in the crowd (highlighted in blue) and Heavily dense crowd (highlighted in yellow). | 1 |
| 1.2 Applications of Crowd Counting in Healthcare (top-left), Urban Planning (top-right), Security (bottom-left) and Surveillance (bottom-right). | 2 |
| 1.3 Datasets of Crowd Counting : The dataset comprises of 1. Images (left) and 2. Point annotations (center) - x,y locations at center of head marked with dots. These Point annotations can be further modified into density maps by applying Gaussian at each point location (right). | 3 |
| 1.4 An overview diagram depicting the employed pipeline of any generic crowd-counting approach. | 3 |
| 1.5 Histogram of the dataset NWPU. The solid orange line represents the zoomed in plot of the head of the distribution which depicts the skew. The dash-dot green line shows the large discontinuities in tail. | 4 |
| 2.1 Histogram of the Large sized dataset, JHU [27] dataset for its Train (Blue), Validation (Red) and Test (Green) splits. The dashed orange box (embedded) indicates the zoomed in plot for the head of the distribution. The solid lavender box (extended below) shows the large discontinuities in the range. Zoomed in plots have been included for all the large sized datasets. | 7 |
| 2.2 Histogram of the Large sized dataset, NWPU [35] dataset for its Train, Validation and Test splits. | 8 |
| 2.3 Histogram of the Large sized dataset, UCF [12] dataset for its Train, Validation and Test splits. | 9 |
| 2.4 Histogram of the Medium sized dataset, ShanghaiTech- Part A (STA) [44] dataset for its Train, Validation and Test splits. | 10 |
| 2.5 Histogram of the Medium sized dataset, ShanghaiTech- Part B (STB) [44] dataset for its Train, Validation and Test splits. | 13 |
| 3.1 The generic crowd counting pipeline is shown. The box highlights the Stages (1,2,3,4) that will be discussed in this Chapter 3. | 14 |
| 3.2 Sample distribution of counts along with examples from the NWPU [35] dataset. The raw data is processed and split into train, validation and test splits. After this step, the data prepared (resized,cropped) for feeding into the network. | 15 |
| 3.3 Fixing Stage-2: Splitting the entire count range into Bayesian Optimal Bins | 17 |
3.4 Notation: A pictorial representation of data and its mathematical notation used in the Section 3.3.

3.5 A candidate partitioning of a subsequence of S ending with the \( r^{th} \) element of S. Finding the optimal partitioning can be thought of as a search over such candidate partitions. Refer to Sec. 3.3.3.

3.6 The posterior probability has two components: 1. Prior over the number of bins \((N_b)\), 2. Likelihood of the partitioning \((P)\).

3.7 The optimal partitioning \( P^* \) lies between the extremes of all the samples in one bin (Left-Single Bin) and each unique count having its own bin (Right-Degenerate Bins).

3.8 Additive Smoothing Demo: A noise sample is being added to the entire range to avoid sparsely spaced bins.

3.9 Fixing Stage-3: Sampling from the minibatches. The generic pipeline involves random sampling of the images from the entire dataset. The modified approach suggests the sampling from the Bayesian Optimal Bins 3.3.

3.10 Minibatching: Round Robin Sampling (RR) and Random Sampling (RS). The numbers on the arrow \( ↓ \) indicate the order of the selection of the bin.

3.11 Bin Loss Function: The figure depicts the ground truth count \( y = 45 \) and the loss function variation with respect to the predicted count \( \hat{y} \). The reference bin is highlighted in dark green. Refer to Sec. 3.5 for details.

3.12 Fixing Stage-4: Strata-aware optimization.

4.1 Scatter plot of the test set errors of DM-Count [34] on NWPU [35] dataset. The Mean Absolute Error (MAE) is 71.71, but the standard deviation is multiple orders of magnitude larger: 376.40. The zoomed in plot shows that even for lowest count (0 people), error is significantly larger than 0 (150). Clearly, MAE is a poor representative of performance across count range.

4.2 Fixing Stage-5: Demo of the inclusive performance measures in comparison to the generic evaluation.

4.3 Example of GAME [10] metric evaluation: The L value used here is 1.

5.1 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over JHU [27] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right. Using these plot, comparative performance for each of the networks across the bin ranges can studied for a given dataset.

5.2 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over NWPU [35] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right.

5.3 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over UCF [12] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right.

5.4 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over STA [44] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right.

5.5 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over STB [44] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right.
5.6 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model P2PNet [28] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Random Sampling - Bin Loss (RS) and Round Robin - Bin Loss (RR) are shown. Using this plot, comparative performance of given Network can be studied over different approaches at a bin level. The table below the plot indicates the bins where the proposed approach performs better than the traditional No-Binning setting. The color code is same as the legend. For example, a green tick mark in the Mean row indicates the mean for the Bin Loss (RR) case is better than the No-binning. .......................... 40

5.7 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model P2PNet [28] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Random Sampling - Bin Loss (RS) and Round Robin - Bin Loss (RR) are shown. Using this plot, comparative performance of given Network can be studied over different approaches at a bin level. .................. 41

5.8 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model FIDTM [20] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. .......................... 42

5.9 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model FIDTM [20] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. .......................... 43

5.10 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model GenLoss [33] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. .......................... 44

5.11 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model GenLoss [33] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. .......................... 45

5.12 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model DM-Count [34] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. .......................... 45

5.13 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model DM-Count [34] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. .......................... 46

5.14 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model BL [24] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. .......................... 46

5.15 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model BL [24] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. .......................... 47
5.16 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SASNet \[29\] over Large datasets JHU \[27\] (top-left), NWPU \[35\] (top-right) and UCF \[12\] (bottom) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. ................................................................. 48

5.17 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SASNet \[29\] over Medium-sized datasets STA \[44\] (left) and STB \[44\] (right) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. ................................................................. 49

5.18 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model S-DCNet \[38\] over Large datasets JHU \[27\] (top-left), NWPU \[35\] (top-right) and UCF \[12\] (bottom) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. ................................................................. 50

5.19 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model S-DCNet \[38\] over Medium-sized datasets STA \[44\] (left) and STB \[44\] (right) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. ................................................................. 51

5.20 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SCAR \[8\] over Large datasets JHU \[27\] (top-left), NWPU \[35\] (top-right) and UCF \[12\] (bottom) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. ................................................................. 52

5.21 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SCAR \[8\] over Medium-sized datasets STA \[44\] (left) and STB \[44\] (right) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. ................................................................. 53

5.22 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SFA-Net \[45\] over Large datasets JHU \[27\] (top-left), NWPU \[35\] (top-right) and UCF \[12\] (bottom) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. ................................................................. 54

5.23 Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SFA-Net \[45\] over Medium-sized datasets STA \[44\] (left) and STB \[44\] (right) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown. ................................................................. 55

5.24 Plot of Percentage Error Threshold (x-axis) and TPER (y-axis) for the different models \[28, 33, 20, 29, 34, 24, 8, 38, 45\] over Large datasets JHU \[27\] (top) and NWPU \[35\] (bottom). Most methods converge to lower error rate other than P2PNet \[28\] and SASNet \[29\]. ................................................................. 55

5.25 Plot of Percentage Error Threshold (x-axis) and TPER (y-axis) for the different models \[28, 33, 20, 29, 34, 24, 8, 38, 45\] over a Large dataset UCF \[12\]. ................................................................. 56

5.26 Plot of Percentage Error Threshold (x-axis) and TPER (y-axis) for the different models \[28, 33, 20, 29, 34, 24, 8, 38, 45\] over Medium-sized datasets STA \[44\] (above) and STB \[44\] (below). Most methods converge to lower error rate other than P2PNet \[28\] and SASNet \[29\]. ................................................................. 56

5.27 Plot between percentage Error threshold (x-axis) and TPER (y-axis) for the network P2PNet \[28\] (left) and FIDTM \[20\] (right) across \[27, 35, 12, 44\] datasets. This plot symbolizes the robustness of the network over different datasets. ................................................................. 57

5.28 Plot between percentage Error threshold (x-axis) and TPER (y-axis) for the network GenLoss \[33\] (top-left), Dm-Count \[34\] (top-right) and BL \[24\] (bottom) across \[27, 35, 12, 44\] datasets. ................................................................. 57
5.29 Plot between percentage Error threshold (x-axis) and TPER (y-axis) for the network SASNet [29](top-left), S-DCNet [38](top-right), SCAR [8](bottom-left) and SFA-Net [45](bottom-right) across [27, 35, 12, 44] datasets. ............................................. 58

6.1 Complete generic and modified pipeline in one place. This diagram depicts the modifications made to cater for the dataset skew (heavy-tail, discontinuity). .......................... 59
List of Tables

| Table | Description | Page |
|-------|-------------|------|
| 2.1   | Summary of the datasets in terms of Year of release, Average Resolution and Number of images. | 6 |
| 2.2   | Summary of the Networks. | 10 |
| 5.1   | Evaluation results on four benchmark datasets JHU, NWPU, UCF-QNRF, ShanghaiTech-A,B (STA,STB) using the evaluation procedure in Chapter 4 on diverse models. The size of test set is indicated below dataset name. The columns represent minibatching schemes (Bin Loss(RS): random bin selection, Bin Loss(RR): round robin bin selection, No-binning: default procedure without binning). For each result, superscript denotes the standard deviation. The best result for each dataset is highlighted in sky blue. The group-wise best is indicated by its group color. The best MAE and standard deviation of the absolute errors are highlighted in bold for each network. Note that first three columns of the table (Pooled MAE and standard deviation) are not directly comparable to the last column of Global MAE and standard deviation values. | 34 |
| 5.2   | Evaluation results in the form of GAME values over four benchmark datasets JHU, NWPU, UCF-QNRF, ShanghaiTech-A,B (STA,STB). The top 3 performing networks are shown in the table below. The columns represent minibatching schemes (Bin Loss(RS): random bin selection, Bin Loss(RR): round robin bin selection, NB: default procedure without binning). The best performer between these three minibatching schemes is bolded (mean and std) network-wise. Group-wise best is indicated by the groups color. | 38 |
| 5.3   | Ablations on the likelihood model for different choices of bin-level distribution. Though the pooled MAE with Poisson distribution is slightly lower for random binning, the standard deviation is significantly larger than our choice (multinomial). | 39 |
Chapter 1

Introduction

In this chapter we will briefly discuss the task of crowd counting, the problem of interest in this thesis, overview of this thesis and publications related to this work.

1.1 Context on Crowd Counting

Figure 1.1: Crowd Counting sample image: Occlusions in the crowd (highlighted in blue) and Heavily dense crowd (highlighted in yellow).
Crowd Counting is the task of counting the number of people in a given image. This task seems rather simpler when the number of people in the given image is less. As the number of people in the image increases, the occlusions increase and the distinction between the people diminishes as shown in the Figure 1.1.

Estimating count people from images has diverse range of applications ranging from security, healthcare, urban planning and surveillance (refer Figure 1.2).

In attempts to solving this problem, the early works included detection based estimation. However, as the density of the crowd increased these detection based approaches ceased to perform. As an alternative, density-based approaches using deep networks were adopted to estimate the crowd count from the image. Deep networks are usually trained on images and ground truth annotations which are either image pixel coordinates (point annotations) of head center of the people or density maps prepared using these coordinates (refer Figure 1.3). These deep networks estimate the total count of people by predicting a density map or point annotations for a given input image. The generic pipeline for crowd counting is shown in Figure 1.4.

During training, Stages 1,2 involve the division of the dataset into train, validation and test splits and preparation of the data into the format that can be fed to the network. Transformations like resizing and cropping are done at this stage. Stage 3 involves the preparation of mini batches using which are fed to
Figure 1.3: Datasets of Crowd Counting: The dataset comprises of 1. Images (left) and 2. Point annotations (center) - x,y locations at center of head marked with dots. These Point annotations can be further modified into density maps by applying Gaussian at each point location (right).

Figure 1.4: An overview diagram depicting the employed pipeline of any generic crowd-counting approach.

A Deep Counting Network. Following this, at Stage 4 an optimization is used to train the network. This optimization varies from approach to approach.

During inference, the evaluation of the approach is performed using Mean Absolute Error (MAE) metric as mentioned in Stage 5 of the Figure 1.4. This error indicates the error between the target and the predicted count from the Deep counting network averaged across all the test images.

1.2 Challenges in Crowd Counting

The customarily used datasets in the crowd counting domain include JHU [27], NWPU [35], UCF [12] and Shanghai Tech-A and B [44]. The datasets are prepared manually by taking an image and marking the center of the head (approximately) for each person in the image. This manual process of dataset preparation is a time-consuming laborious task. The current meticulous datasets have helped the networks improve their performance to a large extent as compared to its predecessor approaches. However, in all of these datasets there is an inherent data-driven insufficiency. This insufficiency originates from challenge in collecting images with higher number of people in the image. For example, the number...
of images with 50-100 people are large in number as compared to images with 10,000-15,000 people. The datasets are all heavy tailed with large discontinuous gaps in the count range as described in the Figure 1.5. This is the dataset skew that we have to deal with while using the crowd counting datasets.

Most crowd counting approaches tend to overlook these characteristics (heavy-tail, discontinuity) of the data distribution and their effect in all stages of the generic crowd counting pipeline (refer Figure 1.4). In the generic pipeline, specifically Stages-1,2 (Dataset preparation and splits) and Stage-3 (Mini-batch Sampling), there is an inherent assumption on the data to follow an approximately uniform data distribution. This assumption fails in this case due to skew (heavy-tail, discontinuity) in the data distribution. The standard optimization techniques (Stage-4) and evaluation techniques (Stage-5) are also oblivious to the skew of the data distribution which makes their loss and error estimations inaccurate.

These issues have been deeply investigated in this work and alternative paradigms that actively factor the data distribution information at each stage of the training pipeline are proposed. To fix the Stages-(1,2) Data preparation and splits, an algorithm has been devised to divide the data range into Bayesian optimal bins (or strata). The Stage-3 is modified to generate the minibatch from the Bayesian optimal strata to ensure strata-level balanced minibatching. The optimization technique (Stage-4) is revised
to include strata-level optimization. The evaluation presented in this thesis introduces three inclusive performance measures to characterise performance. The first alternative evaluation measure includes a strata-level performance evaluation and an aggregation statistic. The second alternative brings in relative error into the picture. This is a novel error built on percentage error and error ratio. The last alternative combines both localization and counting error to select the best approach.

1.3 Overview of the Thesis

The organization of the thesis is as follows:

- The Chapter 2 introduces the background of crowd counting. It mainly highlights the current benchmark datasets and state-of-the-art approaches.

- Chapter 3 launches into the details of the generic crowd counting training pipeline and extends up-to the existing limitations and proposed fixes to overcome these limitations.

- Next, Chapter 4 presents the generic evaluation measures and their shortcoming’s. Improved alternatives measures to the existing evaluation measures are also presented.

- Chapter 5 includes the results of the experiments that provide demonstration of performance change using the modifications discussed in Chapter 3 on benchmark datasets. The results are presented using the same prescription as discussed in the Chapter 4.

- Lastly, Chapter 6 provides conclusion to the existing work and presents the possible direction of progression of this work in the crowd counting domain.

1.4 Publications

Conference:
Sravya Vardhani Shivapuja, Mansi Pradeep Khamkar, Divij Bajaj, Ganesh Ramakrishnan, Ravi Kiran Sarvadevabhatla. Wisdom of (Binned) Crowds: A Bayesian Stratification Paradigm for Crowd Counting. Accepted at ACM Multimedia (ACMMM) 2021. https://dl.acm.org/doi/10.1145/3474085.3475522.

Journal:
Sravya Vardhani Shivapuja, Ashwin Gopinath, Ayush Gupta, Ganesh Ramakrishnan, Ravi Kiran Sarvadevabhatla. Counting in the 2020s: Binned Representations and Inclusive Performance Measures for Deep Crowd Counting Approaches. Under Review at IEEE Transactions on Image Processing. https://arxiv.org/abs/2204.04653.
Chapter 2

Background

To solve the crowd counting problem, there are a few benchmark datasets that provide data. These datasets and their characteristics will be discussed in this chapter. Additionally, related work on different approaches to solve the problem will be dealt in this chapter. Lastly, the limitations of existing work are described to provide ground for the modifications discussed in this thesis.

2.1 Datasets in Crowd Counting

| Size  | Name         | Year | Average Resolution | Number of images |
|-------|--------------|------|--------------------|------------------|
| Large | JHU [27]     | 2020 | 1430 * 910         | 4372             |
|       | NWPU [35]    | 2020 | 2191 * 3209        | 5109             |
|       | UCF [12]     | 2018 | 2013 * 2902        | 1535             |
| Medium| STA [44]     | 2016 | 589 * 868          | 482              |
|       | STB [44]     | 2016 | 768 * 1024         | 716              |

Table 2.1: Summary of the datasets in terms of Year of release, Average Resolution and Number of images.

In crowd counting, the dataset comprises of the image of the crowd and point annotations \(i.e.\) the \(x, y\) pixel location of the center of the person’s head. The crowd counting datasets can be classified into different types free-view [35, 27, 12, 44, 11], surveillance-view [37, 39, 19, 6, 36, 42, 43, 4, 23, 15, 44, 21, 41, 3, 2] and drone-view [30, 46, 1]. In this work, the focus is only on free-view datasets that show a broad variety of images with different orientations of the crowd. Among the free-view datasets, ShanghaiTech [44] (STA,STB), UCF-QNRF [12] (UCF), JHU-CROWD++ [27] (JHU) and NWPU [35] datasets are considered in this thesis. The datasets are divided into two groups into Large Datasets (JHU,
NWPU, UCF) and Medium sized datasets (STA, STB) based on their total number of images as shown in the Table 2.1. The histograms of the Large-sized datasets presented in Figures( 2.1, 2.2, 2.3). These figures indicate the heavy-tailedness (presented in the embedded Orange dashed zoomed in plot) and discontinuity in the tail-end (shown in the lavender solid box below the histogram). This skew propagates throughout the train, validation and test data. Similar trends can be observed in Figures( 2.4, 2.5) for Medium datasets as well.

The dataset skew is famously overlooked when the networks for crowd counting are designed. Most effort has been put into improving the accuracy of predicting count per image, but the data distribution skew is neglected. Let us now look at approaches existing in crowd counting.

Figure 2.1: Histogram of the Large sized dataset, JHU [27] dataset for its Train (Blue) , Validation (Red) and Test (Green) splits. The dashed orange box (embedded) indicates the zoomed in plot for the head of the distribution. The solid lavender box (extended below) shows the large discontinuities in the range. Zoomed in plots have been included for all the large sized datasets.
2.2 Related Work: Deep Networks in Crowd Counting

In this section we will discuss the existing crowd counting approaches. The Section 2.2.1 discusses the training methods and Section 2.2.2 elaborates on the customary evaluation methods.

2.2.1 Training methods

Over the years, there has been tremendous effort put into solving this problem of crowd counting. Few of the significant works are mentioned in this thesis. The earlier works [5, 9, 17] included detection-based approaches which use a detector to identify the people in a given image. These methods do not perform well when there are occlusions in the image. To overcome this constraint, more occlusion-robust approaches based on density were adapted. Further modifications to the density based approaches using the naive point annotations as reference or ground truth are now the state of the art approaches wrt. MAE evaluation. All of these approaches can be broadly clustered into four groups based on the ground-truth and output type of data as shown in 2.2.

The first category of approaches ingests density maps as reference and predicts density maps from which count is estimated (color coded as pink in the Table 2.2). Density-Estimation based approaches are inspired from Zisserman et al. [16] work. In these approaches the point annotations are transformed into density map using Gaussian kernels at each annotation. These density maps are a proxy for the
ground-truth annotation while training a deep-network which regresses the density map for a given input image. One of the earliest attempts to solve this problem was the MCNN [44]. This approach proposes using a multi-column CNN to extract multi-scale features by columns with different receptive field sizes. Zheng et al. [40] introduced a multi-scale feature extractor using different kernel sizes inspired from the Inception module [31]. Sam et al. [25] additionally introduced patch based switching architecture alongside the multi-column CNN to allow selection of the best regressor. Li et al. [18] proposed a well performing- simple approach to enable large receptive fields without brutally increasing the network complexity. Jiang et al. [13] attempts to generate high-quality density maps by encoding spacial hierarchy using Trellis Encoder-Decoder Networks and a multi feature fusion decoder. Liu et al. [22] proposed a network that adaptively encodes multi-level contextual information into the features which regress to the final density map. Latest approaches include [45, 8, 38, 29] which are explained in the Table 2.2.

The second category of approaches deal with point annotations directly on the input side and estimate output density map and crowd count from these maps (indicated by green color in the Table 2.2). The approaches [24, 34, 33] emphasise that the use of point annotations as reference yields accurate results in comparison to density-maps. Jia et al. [32] is another approach along the same train of thought, which models the point annotation noise using a random variable with Gaussian distribution. The approaches [24, 34, 33] are explained in detail in Table 2.2. The third category of approaches include
alternative maps to the classical density map (indicated in yellow in the Table 2.2). FIDT Map [20] is one such alternative. The last category of approaches indicated in purple in Table 2.2 demonstrate a step towards future approaches which are purely point based (P2PNet [28]). Out of all the mentioned approaches, a few approaches have been sampled in each category for experimentation in this thesis. These approaches are explained in the Table 2.2.

Table 2.2: Summary of the Networks.

| Model Name   | Conf., Year | Model Summary                                                                                                                                                                                                                                                                                                                                 | Ground truth          | Output                        |
|--------------|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-------------------------------|
| P2PNet [28]  | ICCV, 2021  | A first-of-a-kind Purely point based framework for crowd counting and localization. This solution predicts a set of point locations consistent with the ground truth point annotations. In order to accomplish the task, the novel idea is to assign learning targets for point proposals which has been implemented using Hungarian algorithm. | Point annotations     | Count estimated from Point Annotations |
| Model | Year | Description |
|-------|------|-------------|
| FIDTM [20] | arXiv, 2021 | To avoid location imprecision and inaccuracy caused by dense overlaps due to Gaussian blurs, this model uses a novel map based on Focal Inverse Distance Transform. |
| GenLoss [33] | CVPR, 2021 | Learning density map representation is modelled as an unbalanced optimal transport problem. A generalized loss is proposed has pixel-wise L2 loss and Bayesian loss [24] as sub-optimal special cases. A perspective-guided cost function is used to deal with the perspective information. |
| DM-Count [34] | NeurIPS, 2020 | Uses Distribution Matching for crowd COUNTing (DM-Count). An Optimal Transport (OT) Loss is used to find the similarity between normalized predicted density map and the normalized ground truth density map. To stabilize this OT loss, a Total Variation loss is used. |
| BL [24] | ICCV, 2019 | To avoid an imperfect “ground-truth” density map which has occlusions, perspective effects, variations in object shapes, etc., this model uses a novel loss function which constructs a density contribution probability model from the point annotations at every pixel. |

Count estimated from Focal Inverse Distance Map |
Count estimated from Density Maps |
Count estimated from Density Maps |
Count estimated from Density Maps |
| Method          | Conference/Journal     | Year | Description                                                                                                                                                                                                 | Count estimated from                  | Density maps constructed from annotations |
|-----------------|------------------------|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------|---------------------------------------------|
| SASNet [29]     | AAAI, 2021             |      | To avoid large scale feature variation problem in crowd counting settings, multi-scale feature representations have been used in multi-level network. Novel PRA loss is introduced to further alleviate the over-estimated or under-estimated region. | Count estimated from Point annotations | Density maps constructed from Point annotations |
| S-DCNet [38]    | ICCV, 2019             |      | Crowd counting task is compared to an open-set problem i.e the count in the image can vary from $[0, +\infty)$. During training however, there is only a smaller closed set observed by the network. Their idea is to build a network that trains on closed set and generalizes well on the open set (unobserved). | Count estimated from Point annotations | Density maps constructed from Point annotations |
| SCAR [8]        | Neurocomputing Journal, 2019 |      | In this work, a Spatial-/Channel-wise Attention Model ramps up the commonly used Regression CNN. The model comprises of Spatial-wise Attention Model (SAM) and Channel-wise Attention Model (CAM). | Count estimated from Point annotations | Density maps constructed from Point annotations |
| SFA-Net [45]    | arXiv, 2019            |      | SFANet uses attention mechanism to generate high quality density maps. It uses VGG model to extract feature map followed by the application of a dual path multiscale fusion network. One of the path retrieves the attention map by extracting the crowd areas and the other path fuses the multi-scale features and attention map to generate high quality density maps. | Count estimated from Point annotations | Density maps constructed from Point annotations |
2.2.2 Evaluation methods

Mean Absolute Error (MAE) and Mean Squared Error (MSE) are the most prevalent evaluation measures in crowd counting approaches. Efforts have been made to examine MAE statistics based on percentage errors, illumination levels and scene levels to characterize performance [35]. However, these are post-hoc measures and do not tackle imbalance which crops up in other stages of the standard pipeline employed for crowd counting.

In this thesis, Thresholded Percentage Error Ratio (TPER) is proposed. The TPER is built on Percentage Error and Error Ratio. Percentage Error is a relative metric that indicates the difference between a predicted value and target value in comparison with the target value. Error Ratio (or Rate) is a measure commonly used for classification setting and is a measure to find the ratio of correctly classified samples to the total number of samples. We extend this measure to a regression setting.

Apart from count errors, there have been extensive measures on the localization performance. P2PNet [28] proposed density normalized Average Precision (nAP), which penalizes both localization errors and false detections. On similar lines, in the vehicle counting domain, Ricardo et al. [10] proposed Grid Average Mean absolute Error (GAME) which simultaneously estimates the object count error and localization error.

In this chapter, we discussed the background of crowd counting exclusively. The datasets existing in crowd counting and their characteristics have been discussed. Detailed approaches and their groups have been mentioned in this chapter. In the next Chapter 3 the generic pipeline of crowd counting will be studied and its shortcomings with solutions will be discussed.
In the last chapter, we had a brief discussion about the existing approaches in crowd counting. In this chapter, we will delve into details of the generic training approaches, highlight their possible limitations and provide systematic solutions.

### 3.1 Current Training Paradigm

Any standard crowd-counting pipeline can be considered as five stages, Data Preparation, Creating Data Splits, Minibatch Creation, Optimization and Evaluation as shown in Figure 1.4. The first four stages comprise of the training pipeline and the last stage is evaluation. This Chapter 3 will focus on the training pipeline and its limitations (Figure 3.1). The following Chapter 4 will discuss the evaluation stage (Stage 5) exclusively.

![Figure 3.1](image)

Figure 3.1: The generic crowd counting pipeline is shown. The box highlights the Stages (1,2,3,4) that will be discussed in this Chapter 3.
3.1.1 Stage-1: Data Preparation

In this stage, the images of crowds and their corresponding point annotations (ground-truth) are subject to transformations like resizing, cropping, density map preparation, etc. These transformations are network specific and transform the raw data into a format that can be utilized by the networks.

Resizing (or down-scaling) is the procedure used on datasets with large resolution images to allow training and evaluation of the networks on moderate-end devices (GPUs with $\sim$ 8-12 GB RAM). Another reason for resizing is to increase the count of people in the crop that is fed to the network. For instance, a crop from the original image will have comparatively lower count compared to the crop from the resized image. Cropping is a must for networks which take fixed size inputs. The size of the crop is set to the size of the input that is ingested by the network. Few approaches require density map as a ground truth. Density map preparation includes translating the point annotations of the heads to a heat-map which represents the density of the people in the image. Majority of the networks use the density map as a proxy to the point annotation.

3.1.2 Stage-2: Creating data splits

The dataset is split into train, validation and test splits as shown in Figure 3.2. These splits are usually assigned by the authors of the dataset itself with a predefined split ratio (e.g., 65%, 15%, 20%). Similar to any training algorithm, the train split is used to train the network. Out of the trained checkpoints, the best performing checkpoint on the validation set is chosen. The test split is used to finally evaluate performance of the network’s checkpoint.

Figure 3.2: Sample distribution of counts along with examples from the NWPU [35] dataset. The raw data is processed and split into train, validation and test splits. After this step, the data prepared (resized,cropped) for feeding into the network.
3.1.3 Stage-3: Minibatch Creation

Following the creation of data splits is the procedure of mini-batching or stacking the data samples up to a pre-defined batch size. The samples are uniformly sampled from the entire dataset with an assumption of uniformly distributed data. This minibatch is sent to the network and the network is trained using the loss on the minibatch. The number of minibatches in an epoch is equal to the \( \frac{\text{Size of the Train Set}}{\text{Batch Size}} \).

3.1.4 Stage-4: Optimization

The parameters of the deep network are optimized for a loss function at minibatch level. Optimization procedures are strategies that adjust the weights to reduce the losses and find the most accurate solutions possible. Different approaches in the crowd counting domain use different optimization procedures based on their network architecture. These optimizations are applied uniformly to all the data samples irrespective of their ground-truth count.

3.2 Limitations of Training pipeline

All of the stages in the training pipeline are developed on certain assumptions on the data. For instance, the Stage 2 (Creating data splits - Sec. 3.1.2) and Stage 3 (Minibatch Creation 3.1.3) assume that the data is uniformly distributed across the count range. However, in crowd counting there is an inherent skew in all the datasets as explained in Section 2.1. In this scenario, uniform sampling of data leads to propagation of the skew into the minibatches leading to under-representation of certain count-ranges (towards the tail end). In the Stage 4 (Optimization-Sec. 3.1.4), the loss function is oblivious to the skew and penalises all the data samples equally putting the least picked samples (towards the tail end) at a disadvantage. To address these issues, the generic pipeline is systematically transformed. Stage 1 (Data Preparation-Sec. 3.1.1) is untouched and modifications are made to remaining stages.

3.3 Fixing Stage-2: Bayesian Optimal Bins

As previously discussed, the standard sampling procedure for creating the train-validation-test splits assumes a uniform distribution which is false in the current setting. The tail end of the distribution is highly under-represented and the network seldom learns on these data samples. The reason behind this effect is the resolution of the sampling. The general sampling procedure occurs at a ground-truth count level. Coarsening this sampling resolution will represent the entire count range better. As an effort towards coarsening the sampling resolution, the count range has been divided into bins (strata) that are optimal for uniform sampling as shown in Figure 3.3.

Formally, let \( N \) be the total number of images and the count range of data is \( R = [0, C] \), where \( C \) is the maximum count in the dataset. The count data \( D \) can be represented in terms of observed
counts $c_i$ in the dataset and their frequencies $f_i$, as $\mathcal{D} = \{ \langle c_i, f_i \rangle \mid i = 1, \ldots, m \}$ (represented pictorially in Figure 3.4), where $m$ is total number of distinct counts in the dataset. Thus, $c_1 = 0, c_m = C$. Consider a partitioning of the counts into $N_b$ bins as:

$$
\mathcal{P}(1, N) \equiv \{ [n_{k-1}, n_k - 1] \}, \quad k = 1, 2, 3 \ldots N_b
$$

(3.1)

where $n_{k-1}$ represents the start index of the $k^{th}$ bin. Note that $n_0 = 0$ and $n_{N_b} - 1 = C$. Here on for the sake of simplicity, the reference to $(1, N)$ when referring to $\mathcal{P}(1, N)$ is dropped.

### 3.3.1 Partition Prior

The prior over partitions is formulated in terms of number of bins $N_b$ in a candidate partition. The prior distribution is denoted as $P(N_b)$. In the attempts of finding partitions that maximize the likelihood the degenerate case of all the unique counts having their own bin might occur. A pictorial representation of this is shown in the Figure 3.7. To avoid this trivial case a constrain over the number of bins is imposed [26]. Specifically, a geometric prior is utilized to assign lower probability to a partition containing larger bin counts:

$$
P(N_b; \gamma) = \begin{cases} 
P_0 \gamma^{N_b} & \text{if } 1 \leq N_b \leq \alpha \\
0 & \text{otherwise} \end{cases}$$

(3.2)
where $P_0$ is a normalization constant. $\gamma < 1$ is a parameter which affects the distribution profile and $\alpha$ controls the practical effectiveness of the upper bound on $N_b$. Applying the laws of probability to $P(N_b)$ and solving for $P_0$ yields in:

$$P(N_b; \gamma) = \frac{1 - \gamma}{1 - \gamma^\alpha} \gamma^{N_b} \tag{3.3}$$

### 3.3.2 Partition Likelihood

The likelihood for a partition $\mathcal{P}$ is defined in terms of the likelihood of each constituent bin in the partition. Let $m_k$ be the width of bin $B_k$. Let the count frequencies of the $m_k$ distinct counts within the bin be denoted by $x_1, x_2, \ldots, x_{m_k}$ respectively. The likelihood for each bin as a multinomial distribution can be modelled as:

$$lik(B_k) = lik(x_1, \ldots, x_{m_k}; p_1, \ldots, p_{m_k}) = \frac{X_k!}{x_1!x_2!\ldots x_{m_k}!} \prod_{j=1}^{m_k} p_j^{x_j} \tag{3.4}$$

where $X_k = \sum_{j=1}^{m_k} x_j$ and $p_j$ is probability of the $j^{th}$ count. Assuming independence across bins, the log likelihood of the partition can be expressed as:
3.3.3 Optimal Partitioning

Given the count range $R = [0, C]$, at one extreme, a partitioning can have just one bin where all the data lies. At the other extreme, a partitioning can have one bin for each unique count in the data. Thus, finding the optimal partitioning can be thought of as a search over candidate partitions that lie between these two extremes (refer Figure 3.7).

To find the optimal partitioning, let us define the Posterior Probability over any given partitioning $\mathcal{P}$:

$$Posterior[\mathcal{P}] \propto Prior[N_b] \times lik[\mathcal{P}]$$  \hspace{1cm} (3.6)

where $N_b$ is the number of Bins (refer Figure 3.6). The optimal partitioning ($\mathcal{P}^*$) maximizes the A Posteriori Score. The MAP can be defined as:

$$MAP[\mathcal{P}^*] = \underset{\mathcal{P}}{\text{argmax}} \log(Prior[N_b]) + \log(lik[\mathcal{P}])$$  \hspace{1cm} (3.7)

To solve this efficiently, a dynamic programming approach [26] is adopted. To begin with, the count frequency data $\mathcal{D}$ is transformed into a sequence of counts $c_1, c_2, \ldots, c_m$ where $c_i$ is repeated $f_i$ times, i.e., $\mathcal{S} = \{c_i, c_1, \ldots, (f_i \text{ times}, 1 \leq i \leq m)\}$. Let $F_{opt}(1, r)$ be the optimal Maximum A Posteriori (MAP) score for the partitioning of a sub-sequence of $\mathcal{S}$ ending with the $r^{th}$ element of $\mathcal{S}$. Following the principle of optimality:

$$lik[\mathcal{P}] = \sum_{k=1}^{N_b} lik(B_k)$$  \hspace{1cm} (3.5)
Figure 3.6: The posterior probability has two components: 1. Prior over the number of bins ($N_b$), 2. Likelihood of the partitioning ($\mathcal{P}$).

$$lik(B_k) = \frac{X_k!}{x_1!x_2!\ldots x_m!} \prod_{j=1}^{m_k} p_j^{x_j}$$

$$P(N_b; \gamma) = \frac{1 - \gamma^N_b}{1 - \gamma^N}$$

Figure 3.7: The optimal partitioning $\mathcal{P}^*$ lies between the extremes of all the samples in one bin (Left-Single Bin) and each unique count having its own bin (Right-Degenerate Bins).
Algorithm 1: Algorithm to find likelihood of a held out subset

\begin{algorithm}
\begin{algorithmic}
\Procedure{FindLikelihood}{$D, ratio, \gamma$}
\State \Comment{Input} Data $D$, train-test split ratio $ratio$, prior param $\gamma$
\State \Comment{Output} Likelihood $lik$ of $D$’s test subset
\State \Comment{Split data into train, test as per $ratio$}
\State $train, test = \text{SplitData}(D, ratio)$
\State \Comment{Find optimal bins using train set (Sec. 3.1.2)}
\State $bins = \text{BayesianOptimalBins}(train, prior = \gamma)$
\State \Comment{Find likelihood of test set wrt optimal bins found earlier (Sec. 3.3.2)}
\State $lik = \text{ComputeBinsLikelihood}(test, bins)$
\EndProcedure
\end{algorithmic}
\end{algorithm}

\begin{equation}
F_{opt}(1, r) = \begin{cases} 
0, & \text{if } r = 1 \\
\max_{1 < j \leq r} \left[ \text{best}(1, j-1) + lik(B_{last})(j, r) \right. \\
\left. + \log P(b_j; \gamma) \right] & \text{if } r = 2, 3, \ldots N
\end{cases}
\end{equation}

where \text{best}(1, j-1) is the memoized (precomputed and stored) best likelihood value (Eqn. 3.5) for the sub-partition ending in the $(j - 1)^{th}$ element, $lik(B_{last})(j, r)$ is the likelihood of the final bin containing the subsequence beginning at the $S$’s $j^{th}$ element and ending with the $r^{th}$ element (see Fig. 3.5). $\log P(b_j; \gamma)$ is the prior on number of bins (Eqn. 3.3). More concretely, $b_j$ is the number of bins that form with $S$’s $j^{th}$ element as the split location for the last bin.

Note that the MAP formulation of $F_{opt}(1, r)$ incorporates the partition likelihood and prior in a Bayesian manner. With respect to the formulation in Eqn (3.8), the optimal set of bins corresponds to the ones obtained for $F_{opt}(1, |S|)$, where $|S|$ is the number of elements in sequence $S$.

3.3.4 Additive Smoothing

The sample distribution in crowd datasets is not only heavy tailed, but also sparse at the tail end. In other words, the distribution is characterized by large count spans which do not have any sample associated with them. This causes the binning procedure described in this section to output a large number of sparsely filled bins. To mitigate this effect, additive smoothing [14] is performed on the data before binning, a demo of the procedure is shown in Figure 3.8. Formally, a smoothing factor $\beta$ is added to each distinct count across the count range $R = [0, C]$. In this case, $\beta = 1$ is set.
Figure 3.8: Additive Smoothing Demo: A noise sample is being added to the entire range to avoid sparsely spaced bins.

Figure 3.9: Fixing Stage-3: Sampling from the minibatches. The generic pipeline involves random sampling of the images from the entire dataset. The modified approach suggests the sampling from the Bayesian Optimal Bins 3.3

3.3.5 Grid-search for optimal hyper-parameters

To determine the optimal set of bins, a grid search is performed with cross-validation over a range of values for (i) distribution profile parameter $\gamma$ (Eqn. 3.2) (ii) the train-validation split ratios. Having determined the optimal hyper-parameter $\gamma_{best}$, the $\gamma_{best}$ is used to find the optimal bins, as outlined in Algorithm 2.
3.4 Fixing Stage-3: Strata based Balanced Minibatch Creation

To subside the effect of the skew induced by the heavy-tailed, discontinuous datasets in crowd counting, the data is optimally stratified as prescribed in Section 3.3. To populate the mini-batch, two types of mini-batching schemes (Figure 3.9) are used. First type is mini-batching called Round-Robin Sampling, involves sampling the bins in a serial fashion until all the bins are sampled. The image is randomly sampled from the selected bin (shown in Figure 3.10). This process is repeated until the entire dataset is accounted for in terms of minibatches which completes one training epoch. This process is followed for every epoch.

Figure 3.10: Minibatching: Round Robin Sampling (RR) and Random Sampling (RS). The numbers on the arrow ↓ indicate the order of the selection of the bin.

Figure 3.11: Bin Loss Function: The figure depicts the ground truth count $y = 45$ and the loss function variation with respect to the predicted count $\hat{y}$. The reference bin is highlighted in dark green. Refer to Sec. 3.5 for details.
In the second type of mini-batching, the bin itself is selected at random and the data sample is picked randomly from this randomly selected bin. Similar to Round Robin Sampling, the sampling process is repeated for the entire dataset over one epoch (in terms of minibatches). Both these sampling methods ensure that the mini-batches are balanced in terms of count range at the resolution of the bins which was not previously the case when uniform sampling was being used. The evaluation between these two methods is done in the Chapter 5.

3.5 Fixing Stage-4: Strata Aware Loss Function

The standard optimization procedure in crowd counting involves calculating the per-sample loss and the averaged loss over the mini-batch is back-propagated. The skew induced by the dataset also propagates the minibatch as mentioned in Section 3.2. Using these minibatches to optimize the network propagates the effect of skew into the optimization procedure as well. This leads to high variance in the absolute error \(|y - \hat{y}|\), where \(y\) is the ground-truth count and \(\hat{y}\) is the predicted count. As an attempt to fix this issue, a novel strata-aware loss function, \(\tilde{L}\) is introduced (refer Figure 3.12). This loss function enables data-distribution aware optimization. The generic optimization depends solely on the absolute error \(|y - \hat{y}|\), this is now modified to incorporate strata. If the predicted count (\(\hat{y}\)) is lying inside the same bin as the ground-truth count (\(y\)) a lower penalty is imposed, else a higher penalty is imposed. The

![Figure 3.12: Fixing Stage-4: Strata-aware optimization](image)

The generic optimization depends solely on the absolute error \(|y - \hat{y}|\), this is now modified to incorporate strata. If the predicted count (\(\hat{y}\)) is lying inside the same bin as the ground-truth count (\(y\)) a lower penalty is imposed, else a higher penalty is imposed. The
lower penalty used here is logarithmic penalty and the higher penalty used is linear penalty as shown in Figure 3.11.

Formally, our strata-aware loss function is defined as:

$$\hat{L} = \begin{cases} 
\lambda_1 \log(1 + |y - \hat{y}|) & \text{if } b_{low} \leq \hat{y} \leq b_{high} \\
|y - \hat{y}| & \text{otherwise}
\end{cases} \quad (3.9)$$

where $b_{low}$ and $b_{high}$ are the bin limits defined by the bin that $y$ belongs to (see Fig. 3.11) and $\lambda_1$ is a weighting factor of the log component. This loss is added as an additive component to the default model loss to encourage strata-aware optimization.

In this chapter we have discussed about the training pipeline and possible modifications that improve the performance. In the next Chapter 4 we will discuss the measures through which these errors will be quantified.
Algorithm 2 Optimal Bins

1: \textbf{procedure} \textsc{optimalBins}(D) \\
2: \hspace{1em} \textbf{▷} \textbf{Input} data D \\
3: \hspace{1em} \textbf{▷} \textbf{Output} Optimal bins \textit{bins}_{\text{best}} \\
4: \hspace{1em} \textbf{▷} \text{Grid search values for } \gamma \text{ (Sec. 3.3.1)} \\
5: \hspace{1em} \Gamma = [0.1, 0.2, \ldots, 0.9] \\
6: \hspace{1em} \textbf{▷} \text{Grid search values for train-test ratios} \\
7: \hspace{1em} \textit{ratios} = [0.1, 0.2, 0.25] \\
8: \hspace{1em} \textbf{▷} \text{Cross-validation repeat factor} \\
9: \hspace{1em} \textit{seeds} = 10 \\
10: \textbf{for} \gamma \text{ in } \Gamma \textbf{ do} \\
11: \hspace{2em} \textbf{for} r \text{ in } \textit{ratios} \textbf{ do} \\
12: \hspace{3em} \textbf{for} f \text{ in } [0:1:\text{seeds}] \textbf{ do} \\
13: \hspace{4em} D_f = \text{shuffle}(D, \text{seed} = f); \\
14: \hspace{4em} \textbf{▷} \text{Algorithm 1} \\
15: \hspace{4em} \textit{lik}_{f,r,\gamma} = \text{findLikelihood}(D_f, r, \gamma) \\
16: \hspace{3em} \textbf{end for} \\
17: \hspace{2em} \textbf{end for} \\
18: \hspace{1em} \textbf{end for} \\
19: \hspace{1em} \textbf{end for} \\
20: \hspace{1em} \textbf{▷} \text{To find the best } \gamma \text{ across all } \textit{ratios}, \text{ descending sort by likelihood for each ratio } r. \\
21: \hspace{1em} \textbf{▷} \text{For each } \gamma, \text{ sum indices of corresponding location in sorted order of earlier step.} \\
22: \textbf{for} \gamma \text{ in } \Gamma \textbf{ do} \\
23: \hspace{2em} \textit{idxsum}_{\gamma} = 0 \\
24: \textbf{for} r \text{ in } \textit{ratios} \textbf{ do} \\
25: \hspace{3em} \textit{idxsum}_{\gamma} + = \text{getDescendingIdx}(\textit{lik}_{r,\gamma}) \\
26: \hspace{2em} \textbf{end for} \\
27: \hspace{1em} \textbf{end for} \\
28: \hspace{1em} \textbf{end for} \\
29: \hspace{1em} \textbf{▷} \text{The best } \gamma \text{ is one with lowest index sum.} \\
30: \gamma_{\text{best}} = \text{argmin} \ \gamma \ \gamma \ \textit{idxsum}_{\gamma} \\
31: \hspace{1em} \textbf{▷} \text{Use the best } \gamma \text{ and determine optimal partitions (Sec. 3.1.2).} \\
32: \textit{bins}_{\text{best}} = \text{BayesianOptimalBins}(D, p_{\text{prior}} = \gamma_{\text{best}}) \\
33: \textbf{end procedure}
Chapter 4

Evaluation measures

This chapter mentions about the existing evaluation methods in Crowd counting and shortcomings as representative performance measures. Alternatively, inclusive performance measures that provide comprehensive picture are introduced.

4.1 Current Evaluation measures and Limitations

In the crowd counting domain the mainly used performance measures are Mean Absolute Error (MAE) and Mean Squared Error (MSE). MAE is the widely accepted and used as a primary performance measure. The formulation of MAE over the test data \( D_{test} \) with \( T \) samples can be defines as Eqn. 4.1.

\[
MAE = \frac{1}{T} \sum_{t=1}^{T} |y_t - \hat{y}_t|
\]  

(4.1)

where \( y_t \) and \( \hat{y}_t \) are the ground-truth and predicted counts for a given test image \( t \). This formulation shows that there is no inclusion of the information of the data distribution in the performance. Observing the per-instance scatter plot of errors in Figure 4.1 tells us that the MAE value does not stand as a representative of the performance due to the high variance of the errors. The deviation of the Absolute Errors on the test set is more than five times the mean error value. This is an effect of the dataset skew explained in Section 2.1. As a consequence of the dataset skew, standard evaluation measures are deemed useless. To overcome this, inclusive performance measures that give insights along different aspects of the performances are presented in this thesis.

4.2 Alternative-1 : Strata-level performance

The first alternative evaluation measure includes reporting the performance at the level of the bin. This provides a range specific performance measure. Additionally, this measure could be used to compare between networks and their performance in the lower and higher count regions. Unlike popular
Figure 4.1: Scatter plot of the test set errors of DM-Count [34] on NWPU [35] dataset. The Mean Absolute Error (MAE) is 71.71, but the standard deviation is multiple orders of magnitude larger: 376.40. The zoomed in plot shows that even for lowest count (0 people), error is significantly larger than 0 (150). Clearly, MAE is a poor representative of performance across count range.

approaches in this domain, the standard deviation alongside the mean value at each bin (refer to Figure 4.2) is also reported. Reporting standard deviation and mean at strata-level gives fine-grained perspective of performance compared to overall MAE. This measure provides an intuitive middle ground between per-image error scatter plots Figure 4.1 and overall MAE. The end-user can use this to get an estimate of performance for the range of counts that their application works in.

If an overall summary statistic is required, the bin-level statistics can be combined to get the result. However, this is not recommended since the metric will have the same shortsightedness as the MAE. The individual bin means and standard deviations can be combined as formulated in the Eqn 4.2 and Eqn 4.3.

\[
\mu_{pool} = \frac{n_1 \mu_1 + n_2 \mu_2 + \ldots + n_b \mu_b}{n_1 + n_2 + \ldots + n_b} \quad (4.2)
\]

\[
\sigma^2_{pool} = \frac{n_1 \sigma^2_1 + n_2 \sigma^2_2 + \ldots + n_b \sigma^2_b}{n_1 + n_2 + \ldots + n_b} \quad (4.3)
\]

4.3 Alternative-2: Thresholded Percentage Error Ratio (TPER)

The measures that are popularly used in crowd counting are MAE and MSE. These measures evade the aspect of relative error. A relative error metric is introduced in this thesis to specify the significance of the error wrt the ground-truth count. The existing relative performance measure Mean Absolute Percentage Error (MAPE) is not a ideal metric in this scenario using the same rationale as the MAE.
Figure 4.2: Fixing Stage-5: Demo of the inclusive performance measures in comparison to the generic evaluation.

As proxy to include relativity, a novel Thresholded Percentage Error Ratio (TPER) is proposed. The formulation for TPER is derived from Percentage Error (PE) and Error Ratio. The PE using the ground-truth count \( y \) and predicted count \( \hat{y} \) is computed as:

\[
PE = \frac{|y - \hat{y}|}{y}
\]  
(4.4)

For the formulation of TPER, the \( \theta \)'s are fixed as linearly increasing values between \([0, 100]\) in steps of 5. The TPER over the test set \((D_{test} \text{ with } T \text{ samples})\) is defined by:

\[
TPER_\theta = \frac{\# \text{ of images for which } PE \geq \theta}{T}
\]

(4.5)

Combining equations Eq. 4.4 and Eq. 4.5, the final form of TPER can be formulated as:

\[
TPER_\theta = \frac{\# \text{ of images for which } (|y - \hat{y}| \geq \theta \times y)}{T}
\]

(4.6)

For a given threshold \( \theta \), \( TPER_\theta \) represents the percentage of test images that have Absolute Error \((|y - \hat{y}|)\) greater than \( \theta \) times the ground-truth count \( y \). The trend of the TPER wrt to the increasing \( \theta \)'s can be depicted graphically as shown in Figure 4.2. The Area Under the Curve (AUC) aids in assessing the overall performance of the networks. A comparative study between the relative error performance can be conducted. The types of comparison studies possible with the TPER curves are:

- **Cross-Network Analysis**: The relative performance analysis across different networks can be analyzed across the threshold (\( \theta \)'s).

- **Cross-Dataset Analysis**: The dataset could be fixed and the analysis across different networks can provide the margin of performance.

- **Cross-Approach Analysis**: Performance can be compared between different modifications of the approaches. One such scenario is comparison between the Binning approaches vs. the Generic Approaches.
4.4 Alternative-3: Grid Average Mean absolute Error (GAME)

In crowd counting approaches, the correct crowd count prediction is equally important as the prediction of the precise location of the head. For instance, if the predicted crowd count is accurate but the location of the people predicted is erroneous. This would imply that the network is not learning the correct targets. To keep the false positives in check, a localization metric (Grid Average Mean absolute Error-GAME [10]) is employed in this thesis. Unlike the common measures of localization which focus only on the localization performance, GAME [10] measures both count error and localization error simultaneously.

![Ground truth map](image1)

![Predicted map](image2)

\[
GAME(L) = \sum_{l=1}^{4^L} |y^l - \hat{y}^l|
\]

Figure 4.3: Example of GAME [10] metric evaluation: The L value used here is 1.

With GAME [10], the image is subdivided into \(4^L\) non-overlapping regions (as shown in Figure 4.3), where \(L \in \mathbb{Z}\), and calculate localized MAE within each sub-regions. Mathematically,

\[
GAME(L) = \frac{1}{T} \sum_{t=1}^{T} \left( \sum_{l=1}^{4^L} |y_t^l - \hat{y}_t^l| \right)
\]

(4.7)

where, \(\hat{y}_t^l\) and \(y_t^l\) are the predicted count and the ground-truth count in a region \(l\) for the image \(t \in T\).

Observably, as \(L\) increases, the number of sub-regions increase exponentially, leading to smaller evaluated regions and higher GAME error.

In this chapter we have discussed about existing evaluation measures and their shortcomings. Additionally, three alternative performance measures are presented which give comprehensive picture of the performance. In the next Chapter 5, the effect of modifications proposed in Chapters 3 and Chapters 4 on the performance will be discussed.
Chapter 5

Experiment and Analysis

5.1 Results

In this chapter, the effect of modifications proposed in the Chapter 3 will be presented in terms of the evaluation measures discussed in Chapter 4. The limitations of MAE (as mentioned in the Chapter 4) create need for more insightful performance metrics. These metrics will be discussed in this chapter. Different sections cater to different statistical measures of performance.

5.1.1 Strata-level statistics

In this section the strata (or bin) level statistics will be discussed. This range specific metric can be used to estimate performance inside and across different count ranges. Full set of interactive Plotly plots are available at the website deepcount.iiit.ac.in as an additional reference.

5.1.1.1 Dataset-wise

Let us look at the performance between different networks for a fixed dataset as shown in Figures( 5.1, 5.2, 5.3, 5.4 and 5.5). All of these Figures indicate the expected trend of increase in mean error and deviation as the ground truth count range increases. Additionally, from these plots the following observations can be drawn.

- From the Figure 5.1, we can observe that the FIDTM [20] and GenLoss [33] show better performance across the count ranges for both RR and RS minibatching. SASNet [29] consistently has bad performance especially in the higher count range bins.

- In Figure 5.2 shows results on NWPU [35] Dataset, P2PNet [28] and SASNet [29] show outlier performance compared to rest of the approaches for NWPU dataset. GenLoss [33] and BL [24] show superior performance towards the higher count bins suggesting that these point-reference-density-map-output approaches could be reliably used for higher count regions.
• Figure 5.3 shows similar trend as Figure 5.2 for P2PNet [28] and SASNet [29]. BL [24], DM-Count [34] and GenLoss [33] show best performance for the UCF [12] dataset across the count ranges.

• In the Figure 5.4 show that all the networks except SASNet [29] show similar performance across the count range for STA [44] dataset.

• Similar to the Figure 5.4, Figure 5.5 shows that approaches except SASNet [29] show reasonably well performance on the STB [44] dataset.

Figure 5.1: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over JHU [27] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right. Using these plot, comparative performance for each of the networks across the bin ranges can studied for a given dataset.

Figure 5.2: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over NWPU [35] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right.
5.1.1.2 Network-wise

In this section we will discuss the Network-wise analysis of the results. This analysis is made from the Figures (5.6, 5.7, 5.8, 5.9, 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16, 5.17, 5.18, 5.19, 5.22, 5.23, 5.20, 5.21). For reducing the visual burden, these figures are embedded with bin-wise Mean and standard deviation performance in terms of tick-marks below the respective bins.

- For P2PNet [28] on all the Large datasets [27, 35, 12] binning helps in improving the performance as shown in Figure 5.6. Medium sized dataset STA [44] and STB [44] do not have the same impact (refer Figure 5.7. This concludes that the purely point based approaches can improve by incorporation of dataset skew.

- The FIDTM [20] network shows significant improvement in performance across the entire count range for JHU [27], UCF [12], STA [44] and STB [44] datasets. On NWPU [35] dataset, performance improvement is observed in the tail-end of the distribution as indicated in Figure 5.8 and Figure 5.9. This behavior indicates that the approach (Chapter 3) discussed in this thesis works for novel focal distance maps (alternative to traditional density maps) as well.

- The point-reference-density-map-output approaches (GenLoss [33], DM-Count [34] and BL [24]) show major performance improvements across all the datasets. This shows that these recent, perspective-changing approaches can utilise the basic dataset skew fixes and benefit in the performance. (refer Figures 5.10, 5.11, 5.12, 5.13, 5.14, 5.15)

- The Figures (5.16, 5.17, 5.18, 5.19, 5.22, 5.23, 5.20, 5.21) show the density-map-reference-density-map-output approaches also show lower errors and deviations when the dataset skew information is utilized to balance the minibatch creation and optimization stages. The improvement is intense for the larger datasets like NWPU [35] and JHU [27].
Figure 5.4: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over STA [44] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right.

5.1.2 Aggregate statistics

This section presents results using aggregate statistics, as mentioned in Chapter 4. The main purpose of this measure is to get an overall-summary statistic for the entire count range from the bin-wise statistics. From the Table 5.1 we can observe that the Bin Loss (RS or RR) (discussed in this thesis) is almost always the best performer for each of the datasets (indicated by cyan). The approach discussed in this thesis performs better 16 out of 20 times. Recent non-traditional approaches like FIDTM [20] and point-reference based approaches like GenLoss [33], DM-Count [34] and BL [24] seem to benefit the most by using the discussed fixes. In these networks the performance improvement is consistent across all the datasets.

Table 5.1: Evaluation results on four benchmark datasets JHU, NWPU, UCF-QNRF, ShanghaiTech-A,B (STA,STB) using the evaluation procedure in Chapter 4 on diverse models. The size of test set is indicated below dataset name. The columns represent minibatching schemes (Bin Loss(RS): random bin selection , Bin Loss(RR): round robin bin selection, No-binning: default procedure without binning). For each result, superscript denotes the standard deviation. The best result for each dataset is highlighted in sky blue. The group-wise best is indicated by its group color. The best MAE and standard deviation of the absolute errors are highlighted in bold for each network. Note that first three columns of the table (Pooled MAE and standard deviation) are not directly comparable to the last column of Global MAE and standard deviation values.

|                |                | Pooled MAE and std | Global MAE and std |
|----------------|----------------|--------------------|--------------------|
|                |                |                    |                    |
| Size     | Dataset | Model   | Bin loss (RS) | Bin loss (RR) | No-binning |
|----------|---------|---------|---------------|---------------|------------|
| JHU[27]  | 1600    | P2PNet  | 204.4±318.2   | 193.2±327.6   | 211.5±365.7| 211.5±538.1|
|          |         | FIDTM   | 71.2±208.5    | 73.0±210.1    | 89.3±278.7 | 89.3±323.6 |
|          |         | GenLoss | 63.5±215.7    | 61.7±211.8    | 61.4±217.5 | 61.4±261.1 |
|          |         | DM-Count| 72.0±224.0    | 69.6±213.3    | 62.1±207.4 | 62.1±244.0 |
|          |         | BL      | 64.3±213.3    | 60.6±210.7    | 63.9±216.4 | 63.9±262.1 |
|          |         | SASNet  | 289.9±117.6   | 292.3±117.6   | 294.7±117.4| 294.7±716.5|
|          |         | S-DCNet | 90.2±225.9    | 86.7±227.9    | 88.1±199.5 | 88.1±313.6 |
|          |         | SCAR    | 78.9±218.9    | 79.1±225.2    | 78.7±223.4 | 78.7±283.2 |
|          |         | SFA-Net | 78.6±255.4    | 82.7±250.1    | 82.5±234.8 | 82.5±277.1 |
| Large    | NWPU[35]| P2PNet  | 379.0±131.4   | 339.8±218.2   | 366.0±160.8 | 366.0±955.4 |
|          | 500     | FIDTM   | 72.0±351.5    | 67.4±390.0    | 69.2±366.4 | 69.2±580.8 |
|          |         | GenLoss | 110.9±322.8   | 107.3±311.6   | 105.2±291.6| 105.2±541.7|
|          |         | DM-Count| 88.1±236.7    | 76.7±205.0    | 77.8±214.9 | 71.7±376.4 [34]|
|          |         | BL      | 112.9±333.7   | 114.8±320.3   | 102.5±348.2| 102.5±560.6 |
|          |         | SASNet  | 355.1±116.0   | 359.5±116.0   | 341.5±115.3| 341.5±954.9 |
|          |         | S-DCNet | 213.4±231.0   | 224.1±230.1   | 210.0±303.1| 248.7±1161.9 |
|          |         | SCAR    | 112.8±321.3   | 111.9±325.6   | 111.3±321.2| 111.3±555.8 |
|          |         | SFA-Net | 136.0±299.1   | 116.4±285.2   | 125.0±343.0| 163.4±1072.1|
| Medium   | UCF[12] | P2PNet  | 535.7±254.5   | 579.5±206.6   | 599.3±264.8 | 599.3±637.0 |
|          | 334     | FIDTM   | 109.7±112.4   | 120.4±150.3   | 215.2±372.4| 215.2±432.5 |
|          |         | GenLoss | 89.5±115.4    | 100.1±109.0   | 95.7±121.0 | 95.7±152.5 |
|          |         | DM-Count| 103.8±107.5   | 97.9±109.1    | 94.5±111.6 | 85.9±120.6 [34]|
|          |         | BL      | 91.1±100.3    | 92.1±105.8    | 98.3±134.2 | 87.1±126.8 [24]|
|          |         | SASNet  | 663.2±146.8   | 631.9±145.6   | 668.5±145.8| 668.5±745.7 |
|          |         | S-DCNet | 205.9±157.8   | 199.2±164.8   | 215.2±190.0| 214.7±277.7 |
|          |         | SCAR    | 124.5±128.6   | 122.9±129.0   | 123.4±146.9| 123.4±197.1 |
|          |         | SFA-Net | 128.6±133.4   | 128.9±162.9   | 128.7±163.2| 128.7±199.9 |
| Medium   | STA[44] | P2PNet  | 67.0±65.5     | 67.3±54.4     | 63.1±54.3 | 58.3±76.7 [28]|
|          | 182     | FIDTM   | 66.3±76.6     | 71.3±84.9     | 68.9±76.4 | 68.9±96.8 |
|          |         | GenLoss | 76.5±73.8     | 69.6±72.2     | 68.9±68.3 | 68.9±93.3 |
|          |         | DM-Count| 88.6±64.4     | 89.6±75.9     | 93.0±81.3 | 64.1±78.4 [34]|
|          |         | BL      | 68.6±69.9     | 68.9±63.3     | 68.7±61.9 | 63.5±74.7 [24]|
|          |         | SASNet  | 365.2±70.2    | 379.9±68.9    | 419.6±66.1| 419.6±352.5 |
|          |         | S-DCNet | 66.6±72.6     | 60.5±65.5     | 61.3±66.9 | 61.3±88.7 |
|          |         | SCAR    | 83.7±67.4     | 72.9±61.8     | 79.3±67.4 | 79.3±82.9 |
### Percentage Error based results

In our experiments we represent the TPER values as curves wrt to the varying thresholds $\theta$. The general trend in all of the plots is Error Ratio decrease with increase in Percentage Error Threshold $\theta$. Since TPER denotes the ratio of test images that have the error greater than $\theta$ times the ground-truth count, this trend is expected. The slope by which the curve declines represents the performance of the network. A steeper curve (lower Area Under the Curve - AUC) indicates lesser images with high relative error and vice versa.

#### Comparative results over different datasets

Different networks can be compared for a given dataset as shown in Figures (5.24, 5.25, 5.26). These TPER curves indicate the comparative performance studies of different networks over a single dataset. The best performing network is denoted by the least Area Under the Curve (AUC). The steeper the curve, better the network. Based on these figures, the following observations can be drawn.

- In the Figure 5.24, for JHU [27] dataset the best performing network is GenLoss [33] followed by DM-Count [34] according to AUC. Most of the networks show similar trends in the TPER curves and are bunched together. The two outlier networks are SASNet [29] and P2PNet [28] which have more than 40% of the test images have Error more than 100 times the ground-truth count.

- In the Figure 5.24, for NWPU [35] dataset the best performing network is GenLoss [33] followed by FIDTM [20] according to AUC. The networks SASNet [29] and P2PNet [28] show similar trend as the JHU dataset.

- In the Figure 5.25, we observe that there is a huge variance between the TPER curves for different networks. This indicates that the networks learn very different for different ranges for the same dataset. SASNet [29] and P2PNet [28] still show high AUCs and which have more than 100% of test images with Percentage Error Threshold ($\theta$) 70.

| Network       | STB [44] | JHU [27] | NWPU [35] |
|---------------|----------|----------|-----------|
| SFA-Net [45] | 68.4±65.1| 64.9±59.5| 63.6±55.6 |
| P2PNet [28]  | 12.0±12.9| 8.8±9.0  | 7.6±8.3   |
| FIDTM [20]   | 8.7±8.6  | 7.8±8.3  | 9.1±10.8  |
| GenLoss [33] | 8.2±9.6  | 9.4±11.9 | 8.8±11.4  |
| DM-Count [34]| 9.1±9.3  | 8.6±8.6  | 8.9±10.3  |
| BL [24]      | 9.6±9.3  | 9.7±9.3  | 10.8±9.2  |
| SASNet [29]  | 66.8±21.2| 67.2±21.1| 67.4±20.6 |
| S-DCNet [38] | 9.2±9.4  | 9.6±10.5 | 7.9±8.6   |
| SCAR [8]     | 9.8±10.2 | 13.8±11.7| 10.3±14.0 |
| SFA-Net [45] | 9.0±7.3  | 8.8±8.0  | 7.4±6.8   |

5.1.3 Percentage Error based results

In our experiments we represent the TPER values as curves wrt to the varying thresholds $\theta$. The general trend in all of the plots is Error Ratio decrease with increase in Percentage Error Threshold $\theta$. Since TPER denotes the ratio of test images that have the error greater than $\theta$ times the ground-truth count, this trend is expected. The slope by which the curve declines represents the performance of the network. A steeper curve (lower Area Under the Curve - AUC) indicates lesser images with high relative error and vice versa.

5.1.3.1 Comparative results over different datasets

Different networks can be compared for a given dataset as shown in Figures (5.24, 5.25, 5.26). These TPER curves indicate the comparative performance studies of different networks over a single dataset. The best performing network is denoted by the least Area Under the Curve (AUC). The steeper the curve, better the network. Based on these figures, the following observations can be drawn.

- In the Figure 5.24, for JHU [27] dataset the best performing network is GenLoss [33] followed by DM-Count [34] according to AUC. Most of the networks show similar trends in the TPER curves and are bunched together. The two outlier networks are SASNet [29] and P2PNet [28] which have more than 40% of the test images have Error more than 100 times the ground-truth count.

- In the Figure 5.24, for NWPU [35] dataset the best performing network is GenLoss [33] followed by FIDTM [20] according to AUC. The networks SASNet [29] and P2PNet [28] show similar trend as the JHU dataset.

- In the Figure 5.25, we observe that there is a huge variance between the TPER curves for different networks. This indicates that the networks learn very different for different ranges for the same dataset. SASNet [29] and P2PNet [28] still show high AUCs and which have more than 100% of test images with Percentage Error Threshold ($\theta$) 70.
Figure 5.5: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over STB [44] dataset for both Round Robin (RR) on the left and Random Sampling (RS) on the right.

- In the next Figure 5.26, for the STA [44] dataset the performance is relatively similar for all the networks except SASNet [29] which shows a high error of 90

- Finally in the same Figure 5.26, for STB [44] dataset all the approaches are performing almost same across the Threshold range except SASNet [29].

5.1.3.2 Robustness of Networks across datasets

TPER curves are useful to compare networks performance across datasets. If a network is robust to different datasets the TPER curves indicating different datasets should be bunched together for that network. The Figures( 5.27, 5.28, 5.29) indicate the robustness or stability of the networks.

The Figure 5.27 shows performance of P2PNet [28] on the left half, from this we can observe that the network P2PNet [28] does not consistently perform across different datasets (shows high variance). This indicates that the purely point based approaches are still rudimentary. On the right half of the figure, FIDTM [20] network shows similar performance for JHU [27] and NWPU [35] datasets. FIDTM [20] network shows that the using novel density-proxy maps could also be a recipe to better performance across datasets.

Figure 5.28 indicate the results of approaches that fall under the class which ingest point annotations from the datasets directly and generate a density map that is used to estimate count. The networks Gen-Loss [33](top-left), DM-Count [34](top-right) and BL [24](bottom) all show similar performance for respective datasets among each other. For each of the plots, the variance is higher than the FIDTM [20] but better than P2PNet [28] from Figure 5.27.

The Figure 5.29 represents the group of network which work with density maps as reference on the input side and as output from the network. There
5.1.4 Localization Error based results

In our experiments, we use GAME [10] as a localization measure. This metric effectively measures both predicted count error and its location. We have experimented with varying $L$ values for the datasets (JHU, NWPU, UCF, STA, STB) and the networks (P2PNet, FIDTM, GenLoss, DM-Count, BL, SASNet, S-DCNet, SCAR, SFA-Net), out of which the top 3 are mentioned in the Table 5.2.

Focal Inverse Distance Map Approach (yellow) is among the top 3 performers in NWPU, UCF, STA and STB. Approaches that ingest point annotations and predict density maps (in green) remains on the top for small as well as large datasets. Overall, the localization performance is improved when Binning (RS and RR) is introduced. There is a performance dip in the group of networks that use density maps for both ground-truth and output (represented in pink). This group performed reasonably well when evaluated by MAE or overall count, but fails when localization performance is considered.

Table 5.2: Evaluation results in the form of GAME values over four benchmark datasets JHU, NWPU, UCF-QNRF, ShanghaiTech-A,B (STA,STB). The top 3 performing networks are shown in the table below. The columns represent minibatching schemes (Bin Loss(RS): random bin selection , Bin Loss(RR): round robin bin selection, NB: default procedure without binning). The best performer between these three minibatching schemes is bolded (mean and std) network-wise. Group-wise best is indicated by the groups color.

| Dataset  | Model                         | GAME metric and std |
|----------|-------------------------------|---------------------|
|          |                               | L=1                | L=2                | L=3                |
| JHU[27]  | GenLoss [33]                  | RS | RR | NB  | RS | RR | NB  | RS | RR | NB  |
| 1455     | 77.4±27.1*                   | 76.2±20.0          | 75.4±27.8          | 89.4±298.7         | 87.9±283.1         | 87.6±285.2         | 107.3±290.9        | 105.3±292.4        |
| NWPU[33] | FIDTM [20]                    | 80.5±561.9         | 77.3±581.3         | 76.2±551.1*        | 88.2±562.0         | 85.6±561.0         | 82.1±531.7         | 101.2±563.8        | 98.6±581.4         |
| 540      | GenLoss [33]                  | 117.3±629.6        | 116.3±625.3        | 114.7±566.1*       | 128.2±630.0        | 128.2±631.7        | 127.8±575.8        | 149.2±625.4        | 149.7±644.9        |
| UCF[12]  | FIDTM [20]                    | 101.1±145.1        | 100.5±136.8        | 104.5±140.0        | 118.5±151.5        | 127.2±151.1        | 124.9±162.0        | 149.2±165.5        | 149.6±165.7        |
| 334      | GenLoss [33]                  | 144.4±145.1        | 145.2±140.0        | 145.5±141.0        | 124.0±162.8        | 121.5±144.1        | 132.8±176.8        | 152.6±168.5        | 153.2±167.8        |
| STA[48]  | FIDTM [20]                    | 78.5±104.9         | 81.2±106.7         | 82.8±101.7         | 91.0±104.2         | 93.4±108.8         | 97.0±103.4*        | 117.0±111.3        | 116.8±113.6        |
| 182      | GenLoss [33]                  | 84.4±93.7          | 84.6±95.4          | 76.7±93.5          | 97.4±106.3         | 91.0±95.1          | 89.6±94.8          | 112.8±99.9         | 118.1±99.9         |
| STB[44]  | FIDTM [20]                    | 11.6±122.1         | 11.7±119.9         | 12.9±114.7         | 15.9±13.5          | 16.6±14.1          | 16.9±15.7          | 22.5±16.8          | 24.0±17.0          |
| 318      | GenLoss [33]                  | 10.4±125.5         | 11.7±140.9         | 10.7±141.4         | 15.0±14.1          | 16.1±16.4          | 14.8±14.9          | 23.7±17.8          | 24.6±19.9          |
|          | DM-Count [34]                 | 11.5±111.8         | 11.1±112.2         | 11.7±112.9         | 15.9±13.0          | 15.4±12.6          | 15.8±13.7          | 23.2±16.3          | 23.0±16.0          |
5.2 Ablations

As mentioned in Sec. 3.3.1, we model the likelihood for each bin as a multinomial distribution. For comparative evaluation, we also consider two other candidate distributions for binning. The first candidate models the likelihood for the bin counts as a Poisson distribution:

\[
lik(B_k) = lik(x_1, \ldots, x_{m_k}; \lambda_1, \ldots, \lambda_{m_k}) = \prod_{j=1}^{m_k} \frac{\lambda_j e^{-\lambda_j}}{x_j} \tag{5.1}
\]

where \(\lambda_1, \ldots, \lambda_{m_k}\) are the parameters of the Poisson distributions associated with the bin elements. The other terms are used in the same context as Eqn. (3.4) in Section 3.3.2. The second candidate distribution for binning is a variant of the multinomial, called stratified multinomial [7]. In this variant, the optimal Bayesian binning is applied not only to the count range, but also to the count frequency distribution. The comparative results can be seen in Table 5.3. Though the pooled MAE with Poisson binning is slightly lower for random binning, the standard deviation is significantly larger than in the case of multinomial (as employed by us). The other results indicate the better overall stability arising from our simple yet effective choice for the likelihood distribution.

Table 5.3: Ablations on the likelihood model for different choices of bin-level distribution. Though the pooled MAE with Poisson distribution is slightly lower for random binning, the standard deviation is significantly larger than our choice (multinomial).

| Likelihood ↓ Binning → | Bin Loss | Bin Loss (RR) | No-binning |
|------------------------|----------|---------------|------------|
| Poisson                | 84.8±441.2 | 89.1±533.1    | 77.8±380.3 |
| Stratified Multinomial | 90.0±283.5 | 90.6±374.0    | 80.7±290.7 |
| Multinomial (ours)     | 88.1±236.7 | 76.7±205.0    | 77.8±214.9 |
Figure 5.6: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model P2PNet [28] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Random Sampling - Bin Loss (RS) and Round Robin - Bin Loss (RR) are shown. Using this plot, comparative performance of given Network can be studied over different approaches at a bin level. The table below the plot indicates the bins where the proposed approach performs better than the traditional No-Binning setting. The color code is same as the legend. For example, a green tick mark in the Mean row indicates the mean for the Bin Loss (RR) case is better than the No-binning.
Figure 5.7: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model P2PNet [28] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Random Sampling - Bin Loss (RS) and Round Robin - Bin Loss (RR) are shown. Using this plot, comparative performance of given Network can be studied over different approaches at a bin level.
Figure 5.8: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model FIDTM [20] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.9: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model FIDTM [20] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.10: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model GenLoss [33] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.11: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model GenLoss [33] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.

Figure 5.12: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model DM-Count [34] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches \textit{i.e.} No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.13: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model DM-Count [34] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.

Figure 5.14: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model BL [24] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.15: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model BL [24] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.16: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SASNet [29] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.17: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SASNet [29] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.18: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model S-DCNet [38] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.19: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model S-DCNet [38] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.20: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SCAR [8] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.21: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SCAR [8] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.22: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SFA-Net [45] over Large datasets JHU [27](top-left), NWPU [35](top-right) and UCF [12](bottom) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.
Figure 5.23: Plot of Strata-level mean and deviation (y-axis) over bin-ranges (x-axis) for the model SFA-Net [45] over Medium-sized datasets STA [44](left) and STB [44](right) across different training approaches i.e. No-Binning, Bin Loss (RS) and Bin Loss (RR) are shown.

Figure 5.24: Plot of Percentage Error Threshold (x-axis) and TPER (y-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over Large datasets JHU [27](top) and NWPU [35](bottom). Most methods converge to lower error rate other than P2PNet [28] and SASNet [29].
Figure 5.25: Plot of Percentage Error Threshold (x-axis) and TPER (y-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over a Large dataset UCF [12].

Figure 5.26: Plot of Percentage Error Threshold (x-axis) and TPER (y-axis) for the different models [28, 33, 20, 29, 34, 24, 8, 38, 45] over Medium-sized datasets STA [44] (above) and STB [44] (below). Most methods converge to lower error rate other than P2PNet [28] and SASNet [29].
Figure 5.27: Plot between percentage Error threshold (x-axis) and TPER (y-axis) for the network P2PNet [28](left) and FIDTM [20](right) across [27, 35, 12, 44] datasets. This plot symbolizes the robustness of the network over different datasets.

Figure 5.28: Plot between percentage Error threshold (x-axis) and TPER (y-axis) for the network GenLoss [33](top-left), Dm-Count [34](top-right) and BL [24](bottom) across [27, 35, 12, 44] datasets.
Figure 5.29: Plot between percentage Error threshold (x-axis) and TPER (y-axis) for the network SAS-Net [29] (top-left), S-DCNet [38] (top-right), SCAR [8] (bottom-left) and SFA-Net [45] (bottom-right) across [27, 35, 12, 44] datasets.
This chapter will walk us through the conclusions of the work discussed in this thesis and future direction that can be taken.

### 6.1 Conclusions

Figure 6.1: Complete generic and modified pipeline in one place. This diagram depicts the modifications made to cater for the dataset skew (heavy-tail, discontinuity).

Throughout the thesis the biases at different stages of the generic crowd counting pipeline are spotlighted. These biases originate due to the data distribution’s skew (heavy-tail, discontinuity). This thesis paves way for the upcoming approaches to incorporate methods that deal with the inherent dataset skew as presented in Figure 6.1. In the training pipeline, the novel Bayesian sample stratification approach...
enables balanced minibatch sampling. Alongside the balanced sampling, novel loss function encourages strata-aware optimization. The proposed bin-aware loss visibly reduces standard deviation of error.

The customarily used evaluation metric, MAE has high standard deviation. This deviation makes MAAE and unreliable performance measure. This thesis accentuates the need of alternative evaluation measures. Making progress towards that direction the evaluation pipeline is upgraded. Inclusive performance measures are presented which give deeper insights about the performance. Firstly, a strata-level performance measure brings range wise insights of the performance. Secondly, a percentage error based metric is used to compare different approaches. Additionally, this metric can be used to compare the robustness of an approach across varying datasets. Lastly, this thesis brings in the localization measure into crowd counting. We hope that this thesis motivates the crowd counting community to understand the severity of the dataset skew and the effects caused by it. We also hope that by studying these challenges and addressing them systematically will lead to statistically reliable crowd counting approaches in future.

6.2 Future Scope

This section will briefly discuss about the directions that the work can take.

- **Generate richer datasets**: A more direct way of solving this problem could be trying to build a dataset that has uniform data distribution across the count range of observation. This would eliminate the dataset skew and perform well on the existing approaches.

- **Stratification at crop level**: As we have discussed in the Section ??, the strata or binning is done at the level of image counts. This is valid direction for network that take resized images and feed it to the network. In few approaches however, the deep counting network trains on the data crops. In this case a stratification at the crop level could be used.

- **Bridge the gap between training and evaluation**: In the some of the approaches in crowd counting, there is a gap between the training and the evaluation procedures. The training pipeline inputs crops of the images to the network while evaluation predicts on the entire image. This might be reason behind high errors in the higher ground-truth count range.

- **Application-specific training**: As an alternative to the Bayesian optimal bins, an application specific binning would help the user. Fine tuning or re-training the existing approaches using the images in its own range might help the network learn better. Evaluation on the application-specific range could give better insights of performance.
Related Publications

**Conference:**
Sravya Vardhani Shivapuja, Mansi Pradeep Khamkar, Divij Bajaj, Ganesh Ramakrishnan, Ravi Kiran Sarvadevabhatla. Wisdom of (Binned) Crowds: A Bayesian Stratification Paradigm for Crowd Counting. Accepted at ACM Multimedia (ACMMM) 2021. [https://dl.acm.org/doi/10.1145/3474085.3475522](https://dl.acm.org/doi/10.1145/3474085.3475522).

**Journal:**
Sravya Vardhani Shivapuja, Ashwin Gopinath, Ayush Gupta, Ganesh Ramakrishnan, Ravi Kiran Sarvadevabhatla. Counting in the 2020s: Binned Representations and Inclusive Performance Measures for Deep Crowd Counting Approaches. Under Review at IEEE Transactions on Image Processing. [https://arxiv.org/abs/2204.04653](https://arxiv.org/abs/2204.04653).
Bibliography

[1] R. Bahmanyar, E. Vig, and P. Reinartz. Mrcnet: Crowd counting and density map estimation in aerial and ground imagery. 2019. 2.1

[2] A. B. Chan, Zhang-Sheng John Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–7, 2008. 2.1

[3] K. Chen, C. C. Loy, S. Gong, and T. Xiang. Feature mining for localised crowd counting. In Proceedings of the British Machine Vision Conference, pages 21.1–21.11. BMVA Press, 2012. 2.1

[4] X. Ding, Z. Lin, F. He, Y. Wang, and Y. Huang. A deeply-recursive convolutional network for crowd counting. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1942–1946, 2018. 2.1

[5] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. IEEE Transactions on Pattern Analysis and Machine Intelligence, 34(4):743–761, 2012. 2.2.1

[6] Y. Fang, B. Zhan, W. Cai, S. Gao, and B. Hu. Locality-constrained spatial transformer network for video crowd counting. arXiv preprint arXiv:1907.07911, 2019. 2.1

[7] J. Florjanczyk and T. Sather. Stratified bayesian blocks. https://medium.com/@janplus/stratified-bayesian-blocks-2bd77c1e6cc7, Aug. 2015. 5.2

[8] J. Gao, Q. Wang, and Y. Yuan. Scar: Spatial-/channel-wise attention regression networks for crowd counting. Neurocomputing, 363:1–8, 2019. (document), 2.2.1, 2.2, 5.1, 5.2, 5.3, 5.4, 5.1, 5.5, 5.20, 5.21, 5.24, 5.25, 5.26, 5.29

[9] W. Ge and R. Collins. Marked point processes for crowd counting. In 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009, 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pages 2913–2920, United States, 2009. IEEE Computer Society. 2009 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009 ; Conference date: 20-06-2009 Through 25-06-2009. 2.2.1

[10] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. López-Sastre, S. Maldonado-Bascón, and D. Oñoro. Extremely overlapping vehicle counting. 06 2015. (document), 2.2.2, 4.4, 4.3, 4.4, 5.1.4

62
[11] H. Idrees, I. Saleemi, C. Seibert, and M. Shah. Multi-source multi-scale counting in extremely dense crowd images. In 2013 IEEE Conference on Computer Vision and Pattern Recognition, pages 2547–2554, 2013. 2.1

[12] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah. Composition loss for counting, density map estimation and localization in dense crowds. In V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, editors, Computer Vision – ECCV 2018, pages 544–559, Cham, 2018. Springer International Publishing. (document), 1.2, ??, 2.1, 2.3, 5.1.1.1, 5.3, 5.1.1.2, 5.1, 5.2, 5.6, 5.8, 5.10, 5.12, 5.14, 5.16, 5.20, 5.22, 5.25, 5.27, 5.28, 5.29

[13] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. S. Doermann, and L. Shao. Crowd counting and density estimation by trellis encoder-decoder networks. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 6126–6135, 2019. 2.2.1

[14] D. Jurafsky and J. H. Martin. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. Prentice Hall PTR, USA, 1st edition, 2000. 3.3.4

[15] D. Kang, D. Dhar, and A. Chan. Incorporating side information by adaptive convolution. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc., 2017. 2.1

[16] V. Lempitsky and A. Zisserman. Learning to count objects in images. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, editors, Advances in Neural Information Processing Systems, volume 23. Curran Associates, Inc., 2010. 2.2.1

[17] M. Li, Z. Zhang, K. Huang, and T. Tan. Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection. In 2008 19th International Conference on Pattern Recognition, pages 1–4, 2008. 2.2.1

[18] Y. Li, X. Zhang, and D. Chen. Csnet: 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. 2.2.1

[19] D. Lian, J. Li, J. Zheng, W. Luo, and S. Gao. Density map regression guided detection network for rgb-d crowd counting and localization. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 2.1

[20] D. Liang, W. Xu, Y. Zhu, and Y. Zhou. Focal inverse distance transform maps for crowd localization and counting in dense crowd. arXiv preprint arXiv:2102.07925, 2021. (document), 2.2.1, 2.2, 5.1.1.1, 5.1, 5.2, 5.3, 5.1.1.2, 5.4, 5.1.2, 5.1, 5.1.3.1, 5.5, 5.1.3.2, 5.1.3.2, 5.2, 5.8, 5.9, 5.24, 5.25, 5.26, 5.27

[21] M. K. Lim, V. J. Kok, C. C. Loy, and C. S. Chan. Crowd saliency detection via global similarity structure. 2014 22nd International Conference on Pattern Recognition, pages 3957–3962, 2014. 2.1

[22] W. Liu, M. Salzmann, and P. Fua. Context-aware crowd counting. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. 2.2.1
[23] M. S. Lu Zhang* and Q. Chen. Crowd counting via scale-adaptive convolutional neural network. 2.1

[24] Z. Ma, X. Wei, X. Hong, and Y. Gong. Bayesian loss for crowd count estimation with point supervision. In Proceedings of the IEEE International Conference on Computer Vision, pages 6142–6151, 2019. (document), 2.2.1, 2.2, 5.1.1.1, 5.1, 5.2, 5.3, 5.1.1.2, 5.4, 5.1.2, 5.1, 5.5, 5.1.3.2, 5.2, 5.14, 5.15, 5.24, 5.25, 5.26, 5.28

[25] D. B. Sam, S. Surya, and R. V. Babu. Switching convolutional neural network for crowd counting. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4031–4039, 2017. 2.2.1

[26] J. D. Scargle, J. P. Norris, B. Jackson, and J. Chiang. Studies in astronomical time series analysis. vi. bayesian block representations. The Astrophysical Journal, 764(2):167, Feb 2013. 3.3.1, 3.3.3

[27] V. Sindagi, R. Yasarla, and V. Patel. Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 1221–1231, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society. (document), 1.2, 2.1, 2.1, 5.1, 5.1.1.2, 5.1, 5.1.3.1, 5.1.3.2, 5.2, 5.6, 5.8, 5.10, 5.12, 5.14, 5.16, 5.18, 5.20, 5.22, 5.24, 5.27, 5.28, 5.29

[28] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Wu. Rethinking counting and localization in crowds: A purely point-based framework. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 3365–3374, October 2021. (document), 2.2.1, 2.2, 2.2.2, 5.1.1.1, 5.1, 5.2, 5.3, 5.1.1.2, 5.4, 5.1, 5.1.3.1, 5.5, 5.1.3.2, 5.1.3.2, 5.6, 5.7, 5.24, 5.25, 5.26, 5.27

[29] Q. Song, C. Wang, Y. Wang, Y. Tai, C. Wang, J. Li, J. Wu, and J. Ma. To choose or to fuse? scale selection for crowd counting. The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21), 2021. (document), 2.2.1, 2.2, 5.1.1.1, 5.1, 5.2, 5.3, 5.4, 5.1, 5.1.3.1, 5.5, 5.1.3.2, 5.5, 5.16, 5.17, 5.24, 5.25, 5.26, 5.29

[30] Y. Sun, B. Cao, P. Zhu, and Q. Hu. Drone-based rgb-infrared cross-modality vehicle detection via uncertainty-aware learning. 2021. 2.1

[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1–9, 2015. 2.2.1

[32] J. Wan and A. Chan. Modeling noisy annotations for crowd counting. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 3386–3396. Curran Associates, Inc., 2020. 2.2.1

[33] J. Wan, Z. Liu, and A. B. Chan. A generalized loss function for crowd counting and localization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 1974–1983, June 2021. (document), 2.2.1, 2.2, 5.1.1.1, 5.1, 5.2, 5.3, 5.1.1.2, 5.4, 5.1.2, 5.1, 5.1.3.1, 5.5, 5.1.3.2, 5.2, 5.10, 5.11, 5.24, 5.25, 5.26, 5.28

[34] B. Wang, H. Liu, D. Samaras, and M. Hoai. Distribution matching for crowd counting. In Advances in Neural Information Processing Systems, 2020. (document), 2.2.1, 2.2, 4.1, 5.1.1.1, 5.1, 5.2, 5.3, 5.1.1.2, 5.4, 5.1.2, 5.1, 5.1.3.1, 5.5, 5.1.3.2, 5.2, 5.12, 5.13, 5.24, 5.25, 5.26, 5.28
[35] Q. Wang, J. Gao, W. Lin, and X. Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020. (document), 1.2, 2.1, 2.2, 2.2.1, 2.2.2, 3.2, 4.1, 5.1.1.1, 5.2, 5.1.1.2, 5.1, 5.1.3.1, 5.1.3.2, 5.2, 5.6, 5.8, 5.10, 5.12, 5.14, 5.16, 5.18, 5.20, 5.22, 5.24, 5.27, 5.28, 5.29

[36] Q. Wang, J. Gao, W. Lin, and Y. Yuan. Learning from synthetic data for crowd counting in the wild. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8198–8207, 2019. 2.1

[37] Q. Wang, D. Hu, L. Mou, J. Gao, Y. Hua, D. Dou, and X. Zhu. Audiovisual crowd counting dataset, May 2020. 2.1

[38] H. Xiong, H. Lu, C. Liu, L. Liang, Z. Cao, and C. Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pages 8362–8371, 2019. (document), 2.2.1, 2.2, 5.1, 5.2, 5.3, 5.4, 5.1, 5.5, 5.18, 5.19, 5.24, 5.25, 5.26, 5.29

[39] Z. Yan, Y. Yuan, W. Zuo, X. Tan, Y. Wang, S. Wen, and E. Ding. Perspective-guided convolution networks for crowd counting. In The IEEE International Conference on Computer Vision (ICCV), October 2019. 2.1

[40] L. Zeng, X. Xu, B. Cai, S. Qiu, and T. Zhang. Multi-scale convolutional neural networks for crowd counting. In 2017 IEEE International Conference on Image Processing (ICIP), pages 465–469, 2017. 2.2.1

[41] C. Zhang, H. Li, X. Wang, and X. Yang. Cross-scene crowd counting via deep convolutional neural networks. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 833–841, 2015. 2.1

[42] Q. Zhang and A. B. Chan. Wide-area crowd counting via ground-plane density maps and multi-view fusion cnns. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, page 8297–8306, 2019. 2.1

[43] Q. Zhang and A. B. Chan. Wide-area crowd counting: Multi-view fusion networks for counting in large scenes. In https://arxiv.org/abs/2012.00946, 2020. 2.1

[44] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 589–597, June 2016. (document), 1.2, ??, 2.1, 2.2.1, 2.4, 2.5, 5.1.1.1, 5.1.1.2, 5.4, 5.1, 5.5, 5.1.3.1, 5.2, 5.7, 5.9, 5.11, 5.13, 5.15, 5.17, 5.19, 5.21, 5.23, 5.26, 5.27, 5.28, 5.29

[45] L. Zhu, Z. Zhao, C. Lu, Y. Lin, Y. Peng, and T. Yao. Dual path multi-scale fusion networks with attention for crowd counting. CoRR, abs/1902.01115, 2019. (document), 2.2.1, 2.2, 5.1, 5.2, 5.3, 5.4, 5.1, 5.5, 5.5, 5.22, 5.23, 5.24, 5.25, 5.26, 5.27, 5.28, 5.29

[46] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling. Detection and tracking meet drones challenge. IEEE Transactions on Pattern Analysis and Machine Intelligence, pages 1–1, 2021. 2.1