Wisdom of (Binned) Crowds: A Bayesian Stratification Paradigm for Crowd Counting

by

Sravya Vardhani Shivapuja, Mansi Pradeep Khamkar, Divij Bajaj, Ganesh Ramakrishnan, Ravi Kiran Sarvadevabhatla

in

ACM International Conference on Multimedia (ACMMM)
: 1
-9

Report No: IIIT/TR/2021/-1

Centre for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
October 2021

Wisdom of (Binned) Crowds: A Bayesian Stratification Paradigm for Crowd Counting

Sravya Vardhani Shivapuja
sravya.vardhani@research.iiit.ac.in
CVIT, IIIT Hyderabad
Hyderabad 500032, INDIA

Mansi Pradeep Khamkar
mansi.khamkar@students.iiit.ac.in
CVIT, IIIT Hyderabad
Hyderabad 500032, INDIA

Divij Bajaj
divij.bajaj.ece17@itbhu.ac.in
CVIT, IIIT Hyderabad
Hyderabad 500032, INDIA

Ganesh Ramakrishnan
ganesh@cse.iitb.ac.in
Dept. of CSE, IIT Bombay
Mumbai 400076, INDIA

Ravi Kiran Sarvadevabhatla
ravi.kiran@iiit.ac.in
CVIT, IIIT Hyderabad
Hyderabad 500032, INDIA

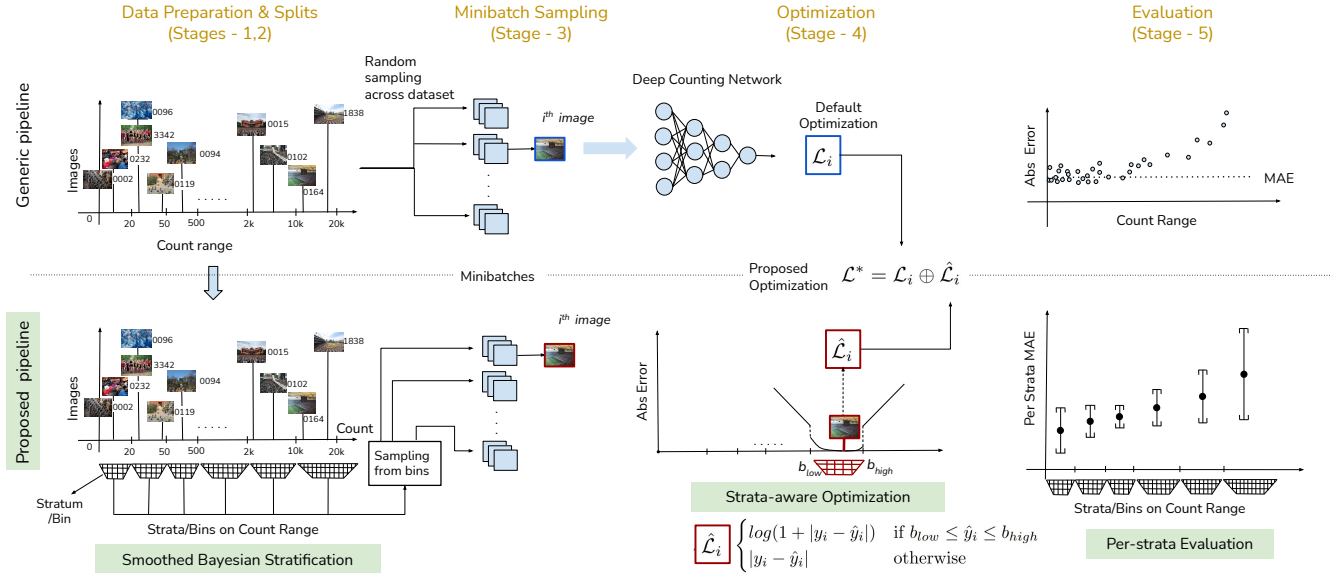

Figure 1: An overview diagram depicting the generally employed processing pipeline of a crowd-counting approach (top) and the proposed modifications we introduce in this work (bottom). See Section 3 for details.

ABSTRACT

Datasets for training crowd counting deep networks are typically heavy-tailed in count distribution and exhibit discontinuities across the count range. As a result, the de facto statistical measures (MSE, MAE) exhibit large variance and tend to be unreliable indicators of performance across the count range. To address these concerns in a holistic manner, we revise processes at various stages of the standard crowd counting pipeline. To enable principled and balanced minibatch sampling, we propose a novel smoothed Bayesian sample stratification approach. We propose a novel cost function which can be readily incorporated into existing crowd counting deep networks to encourage strata-aware optimization. We analyze the performance of representative crowd counting approaches across standard datasets at per strata level and in aggregate. We analyze the performance of crowd counting approaches across standard datasets and demonstrate that our proposed modifications noticeably reduce error standard deviation. Our contributions represent a nuanced, statistically balanced and fine-grained characterization of performance for crowd counting approaches. Code, pretrained models and interactive visualizations can be viewed at our project page deepcount.iiit.ac.in.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475522

CCS CONCEPTS

• **Computing methodologies** → **Scene understanding**.

KEYWORDS

crowd counting, deep network, performance measure

ACM Reference Format:
Sravya Vardhani Shivapuja, Mansi Pradeep Khamkar, Divij Bajaj, Ganesh Ramakrishnan, and Ravi Kiran Sarvadevabhatla. 2021. Wisdom of (Binned) Crowds: A Bayesian Stratification Paradigm for Crowd Counting. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475522

1 INTRODUCTION

Crowd counting is the technique of determining the number of people in a given image. Estimating count from images has significant applications in urban planning, surveillance in industries, hospitals and other establishments. Given an image, deep counting networks regress a single value representing the number of people in the image. Deep networks in crowd counting are typically trained on images and density maps generated from point annotations.

Recent large-scale datasets used to train deep counting networks include Shanghai Tech [14], UCF-QNRF [3] and NWPU-Crowd [11]. Although these datasets have considerably helped advance the state-of-the-art in crowd counting approaches, some issues remain to be addressed. A particularly alarming issue is the heavy-tailed and discontinuous distribution of crowd counts. Specifically, these datasets tend to contain a large number of images with small (people) count and a rather limited number of images with a large count (see Figure 2).

The skew in the data distribution affects all aspects of the problem. It induces imbalance in minibatch sampling, optimization and evaluation. Since the default evaluation protocol (averaging over test errors) does not take the data distribution skew into account, the resulting score (e.g. Mean Absolute Error (MAE)) exhibits high standard deviation, often $2-3$ orders of magnitude higher than MAE itself (see Figure 2). This high deviation prevents mean score from being considered as a reliable performance statistic. Since error deviation is not reported in literature, this issue has gone unaddressed so far.

To address issues mentioned above, we propose an approach that actively factors in the count distribution and its skew at every stage of the problem (see Figure 1). As the first step, we devise an algorithm for partitioning the count range into balanced strata (bins) using Bayesian optimality as a criterion (Sec. 3). The balanced bins form the basis for minibatch sampling (Sec. 3.3). We also formulate a loss function that additionally penalizes error based on the ground-truth binning (Sec. 3.4). Instead of reporting a single performance summary statistic (MAE) across the entire test set range, we report bin-wise statistics and aggregate these statistics in a principled manner (Sec. 3.5) to report the overall score. We perform comparative evaluation involving representative state-of-the-art deep counting networks [2, 7, 10, 12, 15]. Our results (Sec. 5) demonstrate that the proposed approach results in a noticeable reduction of error deviation compared to the default (no-binning) procedure. More generally, our approach helps both designers and end-users determine performance for various count ranges and select from among various approaches based on their relative performance within these ranges.

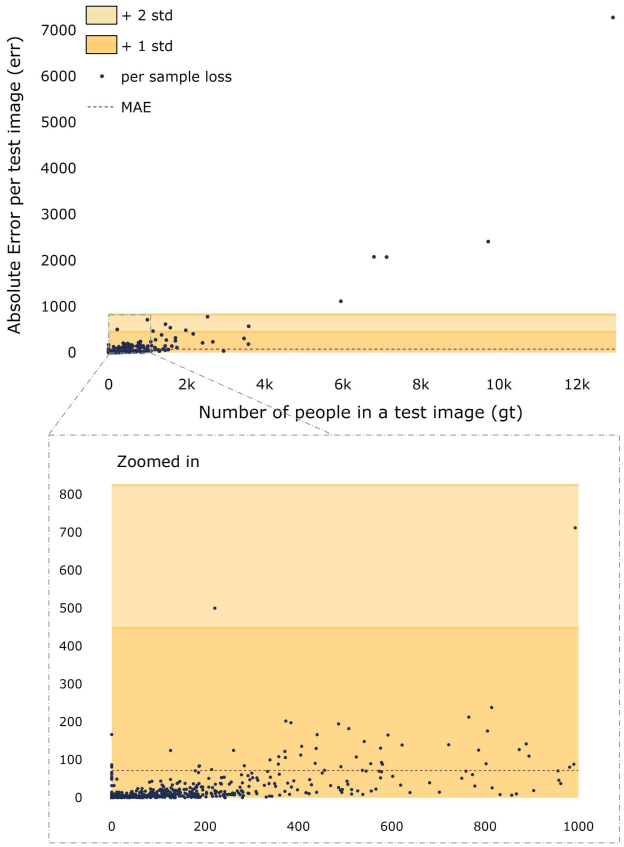

Figure 2: The scatter plot of ground-truth counts and absolute errors by DM-Count [10] on the NWPU dataset [11]. The Mean Absolute Error (MAE) is 71.71, but the standard deviation is multiple orders of magnitude larger: 376.40**. The zoomed in plot shows that even for lowest count (0 people), error is significantly larger than** 0**. Clearly, MAE is a poor representative of performance across count range.**

Code, pretrained models and visualizations can be accessed from our project page deepcount.iiit.ac.in.

2 RELATED WORK

To the best of our knowledge, no works have analyzed the processing pipeline for crowd counting in entirety. In this section, we review works which aim to address some aspects raised in the earlier section.

Density-based crowd counting: Deep Convolutional Networks which represent the target count as a density map form the most popular class of approaches [5, 6, 13, 14]. Some approaches have attempted to address count distribution imbalance, although in an indirect manner. Sam *et al.* [8] propose a switching CNN based model which employs three regressors and a classifier which selects the best regressor to which an input patch is to be routed. There have also been attempts at reducing the skew at the patch level as in Xiong *et*

al. [12]. They discretize the count range into a set of intervals and design a classifier on these intervals, thereby converting an open set regression problem to a closed set classification one.

Point-based crowd counting: To overcome the performance sensitivity to density map preparation, recent approaches use point annotations directly to estimate count. Ma *et. al.* [7] use a novel loss function that constructs a density distribution indirectly from the point annotations. Wang *et al.* [10] employ the optimal transport (OT) loss to find similarity between predicted density map values and ground truth binary point map and a total variation loss to stabilize the OT computation.

Evaluation methods: Mean Absolute Error (MAE) and Mean Squared Error (MSE) are the most prevalent evaluation measures in crowd counting approaches, with MAE usually being the more direct measure. More recently, some attempts have been made to examine MAE statistics based on percentage errors, illumination levels and scene levels to characterize performance [11]. 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.

3 PROPOSED METHOD

3.1 Standard Processing Pipeline

As depicted in Figure 1, any standard approach to crowd-counting can be considered to have five stages:

- *Stage-1 (Data preparation):* In this stage, images and corresponding counts are processed suitably and are provided as input and output to a reference deep network. This stage includes standard procedures such as image cropping and resizing, density map preparation, *etc.*
- *Stage-2 (Creating data splits):* The prepared data is partitioned into training, validation and test splits according to a predefined split ratio (*e.g.*, 65%, 15%, 20%).
- *Stage-3 (Minibatch creation):* The deep network is trained using a subset of data randomly sampled from the training set, usually referred as a minibatch. The training set is partitioned into minibatches for each training epoch.
- *Stage-4 (Optimization):* The parameters of the deep network are optimized for a loss function at the minibatch level.
- *Stage-5 (Evaluation):* A standard performance measure (*e.g.*, MAE) is used for evaluating the model on the validation or the test set.

Each of these stages involves a set of assumptions which are often implicit. For instance, the train-validation-test splitting (Stage-2) and minibatch creation (Stage-3) assume that the distribution over the targets (counts) is uniform. However, target distributions for standard crowd counting datasets are heavy-tailed. Due to the uniform nature of sampling, the data splits and consequently, the training minibatches, exhibit the same heavy-tailed distribution. This skew induces a bias which penalizes samples in the tail during optimization (Stage-4). Due to this bias, the statistical summary measures (*e.g.*, MSE, MAE) fail as representative measures of performance (Stage-5).

To address these issues, we revisit the entire problem setting and propose alternative paradigms for the stages mentioned previously. We leave Stage-1 untouched and describe our modifications to the subsequent stages.

3.2 Revisiting Stage 2 (Creating Data splits)

As mentioned earlier, the standard sampling procedure for creating train-validation-test splits implicitly assumes a uniform distribution over the target range. However, doing so causes the tail portion of the distribution to be under-represented. A fundamental reason for this effect is that the sampling is conducted at too fine a resolution, *i.e.* at the level of individual counts.

One approach to address this issue is to coarsen the resolution and partition the count range into bins (strata) that are optimal for uniform sampling. Formally, let the total number of images be N and suppose the count range over the data samples is $R = [0, C]$, where C is the maximum crowd count. The count data $\mathcal{D}$ can be represented in terms of observed discrete counts c_i and their frequencies f_i, as $\mathcal{D} = \{\langle c_i, f_i \rangle \mid i = 1, ...m\}$, 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:

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

where n_{k-1} represents the start index of the k^{th} bin. Note that $n_0 = 0$ and $n_{N_b} - 1 = C$. For simplicity, we drop the reference to $(1, N)$ when referring to $\mathscr{P}(1, N)$ in what follows.

3.2.1 Partition Prior. We formulate the prior over partitions in terms of number of bins N_b in a candidate partition. In what follows, we refer to this prior distribution as $P(N_b)$. To avoid the degenerate case in which each unique count in the range might land up in its own bin, we impose constraints over the number of bins [9]. Specifically, we use a geometric prior to assign lower probability to a partition containing larger bin counts:

$$P(N_b; \gamma) = \begin{cases} P_0 \ \gamma^{N_b} & \text{if } 1 \leqslant N_b \leqslant \alpha \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

where P_0 is a normalization constant. $\gamma < 1$ is a parameter which affects the distribution profile and α 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, we obtain:

$$P(N_b; \gamma) = \frac{1 - \gamma}{1 - \gamma^{\alpha}} \gamma^{N_b} \tag{3}$$

3.2.2 Partition Likelihood. The likelihood for a partition $\mathscr{P}$ 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. We model the likelihood for each bin as a multinomial distribution:

$$\begin{aligned} lik(B_k) &= lik(x_1, \ldots, x_{m_k}; p_1, \ldots, p_{m_k}) \\ &= \frac{X_k!}{x_1! x_2! \ldots x_{m_k}!} \prod_{j=1}^{m_k} p_j^{x_j} \end{aligned} \tag{4}$$

where $X_k = \sum_{j=1}^{m_k} x_j$ and p_j is probability of the j^{th} count. Assuming bin-level independence, the *log* likelihood of the partition can be expressed as:

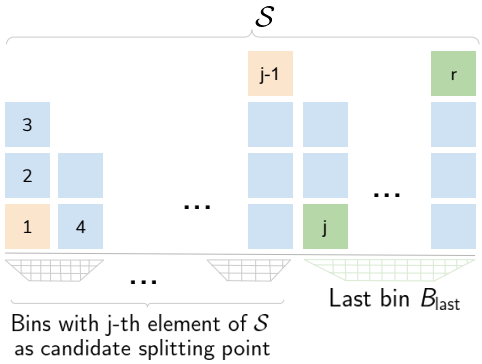

Figure 3: A candidate partitioning of a subsequence of $\mathcal{S}$ ending with the r^{th} element of $\mathcal{S}$. Finding the optimal partitioning can be thought of as a search over such candidate partitions. Refer to Sec. 3.2.3.

$$lik[\mathscr{P}] = \sum_{k=1}^{N_b} lik(B_k) \tag{5}$$

3.2.3 Optimal Partitioning. Given the count range $R = [0, C]$, at one extreme, we can have a partitioning wherein all data lies in a single bin. At the other extreme, we can have a partitioning wherein each unique integer in the range R is a bin. Thus, finding the optimal partitioning can be thought of as a search over candidate partitions that lie between these two extremes.

To solve this task efficiently, we adopt a dynamic programming approach [9]. To begin with, we transform the count frequency data $\mathcal{D}$ into a sequence of counts $c_1, c_2 ....c_m$ where c_i is repeated f_i times, *i.e.*, $\mathcal{S} : \{c_i, c_i, \ldots (f_i \text{ times}, 1 \leqslant i \leqslant m)\}$. Let $\mathcal{F}_{opt}(1, r)$ be the optimal Maximum A Potseriori (MAP) score for the partitioning of a subsequence of $\mathcal{S}$ ending with the r^{th} element of $\mathcal{S}$. Following the principle of optimality, we have:

$$\mathcal{F}_{opt}(1,r) = \begin{cases} 0, \text{ if } r = 1 \\ \max\limits_{1<j\leqslant r} \Big[\texttt{best(1,j-1)} + lik(B_{last})(j,r) \\ \qquad + log\ P(b_j;\gamma) \Big] \text{ if r=2,3...N} \end{cases} \tag{6}$$

where `best(1,j-1)` is the memoized (precomputed and stored) best likelihood value (Eqn. 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 $\mathcal{S}$'s j^{th} element and ending with the r^{th} element (see Fig. 3). $log\ P(b_j;\gamma)$ is the prior on number of bins (Eqn. 3). More concretely, b_j is the number of bins that form with $\mathcal{S}$'s j^{th} element as the split location for the last bin.

Note that the MAP formulation of $\mathcal{F}_{opt}(1,r)$ incorporates the partition likelihood and prior in a Bayesian manner. With respect to the formulation in Eqn 6, the optimal set of bins corresponds to the ones obtained for $\mathcal{F}_{opt}(1, |\mathcal{S}|)$, where $|\mathcal{S}|$ is the number of elements in sequence $\mathcal{S}$.

3.2.4 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, we perform additive

Algorithm 1 Optimal Bins

procedure OPTIMALBINS($\mathcal{D}$)
  ▷ **Input** data $\mathcal{D}$
  ▷ **Output** Optimal bins $bins_{best}$
  ▷ Grid search values for γ (Sec. 3.2.1)
  $\Gamma = [0.1, 0.2, \ldots 0.9]$
  ▷ Grid search values for train-test ratios
  $ratios = [0.1, 0.2, 0.25]$
  ▷ Cross-validation repeat factor
  $seeds = 10$
  for γ in Γ **do**
    for r in $ratios$ **do**
      for f in $[0 : 1 : seeds]$ **do**
        $\mathcal{D}_f$ = shuffle($\mathcal{D}$,$seed = f$);
        ▷ Algorithm 2
        $lik_{f,r,\gamma}$ = FINDLIKELIHOOD($\mathcal{D}_f, r, \gamma$)
      end for
      ▷ Compute average likelihood for a fixed γ and r
      $lik_{r,\gamma}$ = MEAN($lik_{f,r,\gamma}$)
    end for
  end for
  ▷ To find the best γ across all $ratios$,
  ▷ descending sort by likelihood for each ratio r.
  ▷ For each γ, sum indices of corresponding location
  ▷ in sorted order of earlier step.
  for γ in Γ **do**
    $idxsum_\gamma = 0$
    for r in $ratios$ **do**
      $idxsum_\gamma$+ = GETDESCENDINGINDX($lik_{r,\gamma}$)
    end for
  end for
  ▷ The best γ is one with lowest index sum.
  $\gamma_{best} = \underset{\gamma}{\text{argmin}}\ idxsum_\gamma$
  ▷ Use the best γ and determine optimal partitions (Sec. 3.2).
  $bins_{best}$ = BAYESIANOPTIMALBINS($\mathcal{D}, prior = \gamma_{best}$)
end procedure

Algorithm 2 Algorithm to find likelihood of a held out subset

procedure FINDLIKELIHOOD($\mathcal{D}, ratio, \gamma$)
  ▷ **Input** Data $\mathcal{D}$,train-test split ratio $ratio$, prior param γ
  ▷ **Output** Likelihood lik of $\mathcal{D}$'s test subset
  ▷ Split data into train, test as per $ratio$
  $train$, $test$ = SPLITDATA($\mathcal{D}, ratio$)
  ▷ Find optimal bins using train set (Sec. 3.2)
  $bins$ = BAYESIANOPTIMALBINS($train, prior = \gamma$)
  ▷ Find likelihood of test set
  ▷ wrt optimal bins found earlier (Sec. 3.2.2)
  lik = COMPUTEBINSLKHOOD($test, bins$)
end procedure

smoothing [4] on the data before binning. Formally, a smoothing factor β is added to each distinct count across the count range $R = [0, C]$. In our case, $\beta = 1$.

3.2.5 Grid-search for optimal hyperparameters. To determine the optimal set of bins, we first perform a grid search with cross-validation over a range of values for (i) distribution profile parameter γ (Eqn. 2) (ii) the train-validation split ratios. Having determined the optimal hyperparameter γ_{best}, we utilize the same to obtain the optimal set of bins, as outlined in Algorithm 1.

3.3 Revisiting Stage 3: Minibatch Creation

To address the skew induced by the heavy-tailed, discontinuous count distribution of data samples, we bin the data optimally using the procedure described in Section 3.2. To populate a minibatch using our Round Robin (RR) method, we pick a data sample randomly from each of the bins in a round robin fashion, beginning at the first bin. This process is repeated until all the bins have been selected or the minibatch is full. We continue this process until the entire training dataset is accounted for as an epoch (*i.e.*, in terms of minibatches). This procedure is followed for each epoch.

Another variant of binning which we consider is Random Sampling (RS) procedure where a bin is first picked randomly from available bins and a data sample is picked randomly from the randomly selected bin. A procedure similar to Round Robin (RR) is used to populate an epoch's equivalent of training data. Effectively, both our procedures ensure that the mini-batches are balanced in terms of their count range unlike the standard random shuffle-based approach. We analyze the results on both the binning strategies during evaluation (Sec. 5).

3.4 Revisiting Stage 4: Optimization

The standard protocol for optimizing a deep counting network is to minimize the per-instance loss averaged over the minibatch. However, one is confronted with the same issues (imbalance, bias) as those faced during minibatch creation (Sec. 3.3). As a consequence, the trained networks exhibit high variance for the error term $|y - \hat{y}|$, where y is the ground-truth count and $\hat{y}$ is the predicted count.

To enable data-distribution aware optimization, we introduce a novel bin sensitive loss function $\widehat{\mathcal{L}}$. Instead of the loss depending solely on the error, we also consider the count bin to which the data sample belongs and whether the predicted count $\hat{y}$ lies within this bin or outside it. If $\hat{y}$ lies within the bin, we impose a smaller logarithmic penalty. If the count value lies outside, we impose a linear penalty. Formally, our strata-aware loss function is defined as:

$$\widehat{\mathcal{L}} = \begin{cases} \lambda_1 \, log(1 + |y - \hat{y}|) & \text{if } b_{low} \leqslant \hat{y} \leqslant b_{high} \\ |y - \hat{y}| & \text{otherwise} \end{cases} \tag{7}$$

where b_{low} and b_{high} are defined by the bin that y belongs to (see Fig. 4) and λ_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.

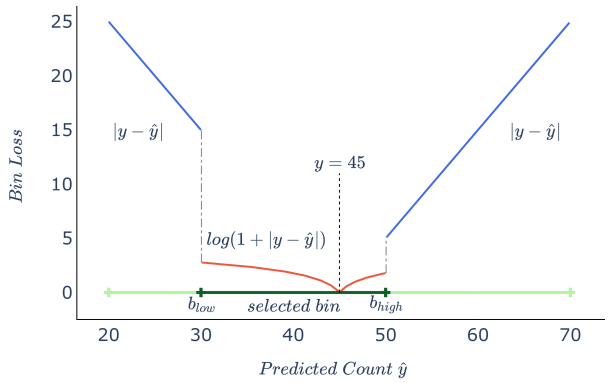

Figure 4: Bin Loss Function : The figure depicts the ground truth count $y = 45$ and the loss function variation with respect to the predicted count $\hat{y}$ inside the bin ($log(1 + |y - \hat{y}|)$) and outside ($|y - \hat{y}|$). The reference bin is highlighted in dark green. Refer to Sec. 3.4 for details.

3.5 Revisiting Stage 5: Evaluation

The discontinuous and heavy-tailed distribution of samples affects the evaluation stage as well. Coupled with lack of bin-level awareness during optimization, an outlier effect arises which causes the default measures (*e.g.*, MSE, MAE) to be ineffective representatives of performance *across* the entire count range. Even more worryingly, the standard deviation of error tends to be at the same level as the mean statistic. Instead of using a single pair of numbers (mean, standard deviation) to characterize performance across the entire count range, we make the following proposals.

One, the evaluation measure must be reported at the level of each bin. This provides a more comprehensive picture of performance. Additionally, it also helps compare the relative effectiveness of various counting networks for smaller and larger counts. *Two,* even if an overall summary statistic over the test set is deemed necessary, the mean and standard deviation of bin-level performance measures are combined in a statistically sound manner. Let the mean and standard deviations for the individual bins be $(\mu_i, \sigma_i); i = 1, 2, \ldots N_b$ and let the number of samples in each bin be n_i. We compute the pooled mean and standard deviation as their weighted average:

$$\mu_{pool} = \frac{n_1\mu_1 + n_2\mu_2 + \ldots + n_{N_b}\mu_{N_b}}{n_1 + n_2 + \ldots + n_{N_b}} \tag{8}$$

$$\sigma^2_{pool} = \frac{n_1\sigma_1^2 + n_2\sigma_2^2 + \ldots + n_{N_b}\sigma^2_{N_b}}{n_1 + n_2 + \ldots + n_{N_b}} \tag{9}$$

4 EXPERIMENTAL SETUP

We perform experiments with two large-scale crowd counting datasets NWPU [11] and UCF-QNRF [3] as well as two variants of the medium-scale dataset ShanghaiTech(A,B) [14]. Although we revisit all stages of the problem pipeline, we retain the standard train and test datasets for consistency. To determine optimal bin hyperparameters (Section 3.2), we isolate a random 20% subset of the train set and use the same for validation. Since NWPU's test set is not directly available, we use the publicly available validation

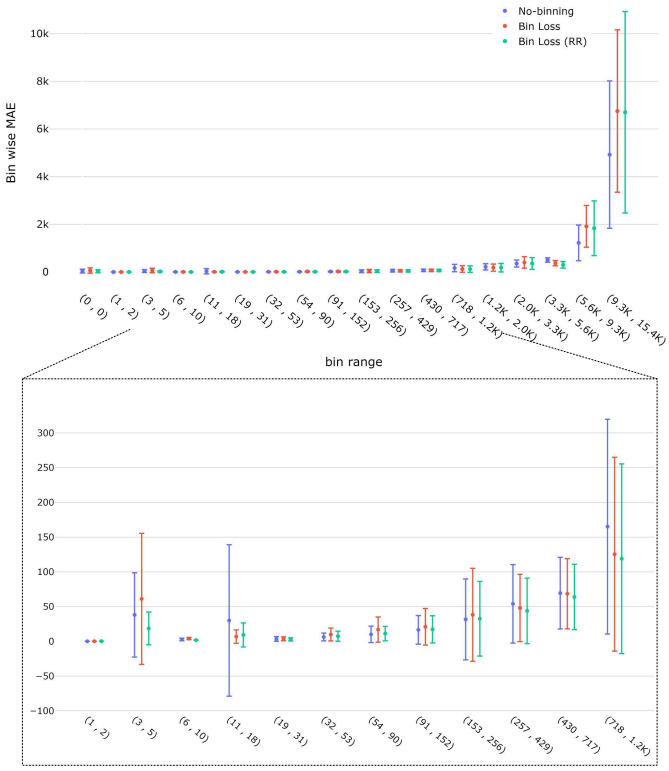

Figure 5: Per-bin performance of DM-Count [10] on NWPU dataset [11] for different binning schemes (color-coded). MAE is represented by a dot and error bars represent standard deviation. Bins in range $[1, 1.2k]$ are shown zoomed in for better visibility. The comparatively larger deviations for the no-binning scheme are clearly evident.

set as the test set and report results on the same. We also compare the two different binning schemes mentioned in Section. 3.3, *viz.*, round-robin (RR) and random selection (RS). For evaluation, we utilize representative and recent state-of-the-art crowd counting networks, *viz.*, DM-Count [10], Bayesian Crowd Counting (BL) [7], SCAR [2], SFA-Net [15], S-DCNet [12]. These papers report results on the ShanghaiTech and UCF-QNRF datasets but not on NWPU (except for DM-Count). Therefore, we report respective test set results by training these networks on the NWPU dataset as well.

The network architecture, ground truth generation, augmentation and image pre-processing steps are used as mentioned in the respective works. We use the hyperparameters, optimizers and loss functions used as suggested in the original implementations of the networks. As mentioned previously, we add the bin-aware loss function (Sec. 3.4) to the original loss function used by the models during optimization. We compute the per-bin MAE and associated standard deviation. We also aggregate the resulting statistics to obtain an overall performance score across the bins (Sec. 3.5). Although not directly comparable to our proposed performance score, we also report the standard MAE (which does not involve any binning) as computed by existing works. As a new addition, we

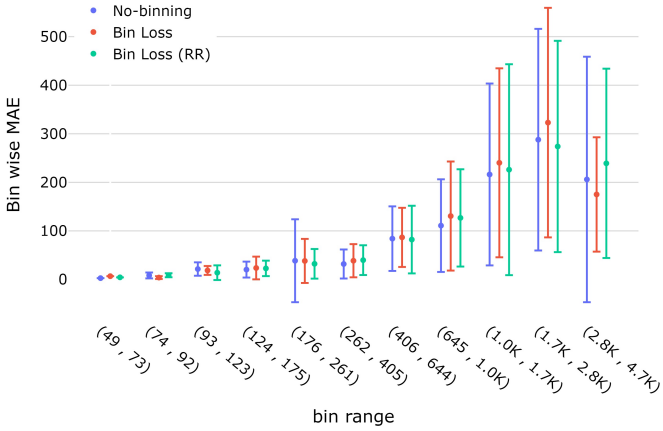

Figure 6: Per-bin performance of DM-Count on UCF dataset. The comparatively larger deviations for no-binning scheme are clearly evident as with other plots.

also report the error's standard deviation. For baseline comparison, we also train models using the default (no-binning) procedure and without the bin-aware loss function included.

5 RESULTS

5.1 Bin-level results

The bin-level mean error scores and the corresponding standard deviation bars can be viewed for a selection of different datasets and binning schemes in Figures 5, 6, 7 and 8. The comparatively large deviations typically incurred when binning is not used can clearly be seen. Also note that the bin-level plots provide a larger perspective on the performance of the approach across the count range, in contrast to a single number which is usually reported. Our project page deepcount.iiit.ac.in contains interactive visualizations for examining results on a per-dataset and per-model (approach) basis.

5.2 Aggregate results

The aggregate scores (described in Section 3.5) can be viewed in Table 1 – refer to the three gray-shaded columns. Across networks and datasets, a reduction in error standard deviation is clearly apparent when bin-aware loss is used (relative to the no-binning counterpart). The aggregate scores reinforce the trend seen in the bin-level plots discussed previously. The reduction in standard deviation compensates for the marginally inferior mean score (compared to no-binning) in some cases. As the blue highlighted results in Table 1 indicate, binning schemes provide the best overall aggregate results across the datasets (except for the smaller count STB dataset).

In the last column of Table 1, we also present the usually reported MAE measure. The results using models made available by authors are indicated. For the first time, we also report the standard deviation for the sake of completeness and consistency. Note that the numbers in this column are not directly comparable with other (gray) columns of the table due to the significant differences

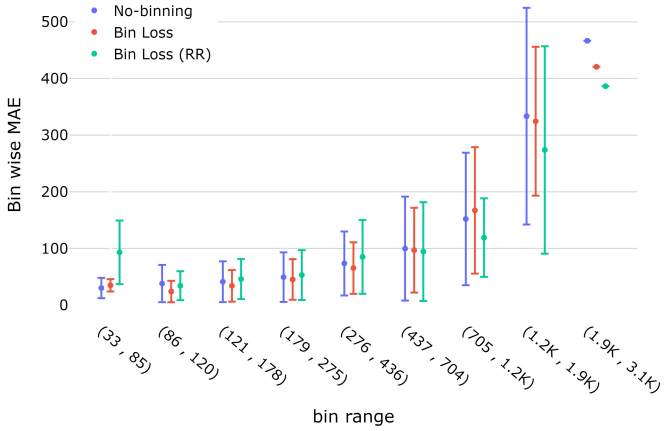

Figure 7: Per-bin performance of DM-Count on STA dataset. Similar to our observation in the earlier plots, the comparatively larger deviations for the no-binning scheme are clearly evident.

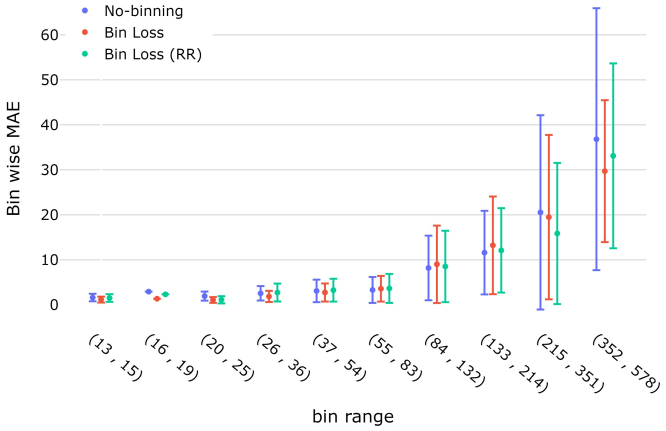

Figure 8: Per-bin performance of DM-Count on STB dataset. The comparatively larger deviations for the no-binning scheme are clearly evident, like in the earlier plots.

across the processing pipeline stages. However, the magnitude of the deviation incurred even by the state of the art approaches is somewhat alarming. It is also interesting to note that the MAE performance ranking for different networks differs significantly from the binning (Pooled MAE) results. For instance, BL [7] is the best performer on UCF with Pooled MAE. A similar trend can be seen for the STA and STB datasets as well. Due to unavailability of BL-specific settings for NWPU dataset, we used the settings used for BL with UCF-QNRF. These settings may be sub-optimal and might be the reason BL underperforms on NWPU.

In our experiments, we tried two minibatching schemes (balanced, random) to determine their effect on performance, if any (Section 3.3). The aggregate results across datasets suggests that random sampling has better overall performance approximately half the time (Table 1, first two columns). Also, the results suggest that random sampling of bins works best for top performing networks (DM-Count [10], BL [7]) most of the time.

5.3 Ablation Studies

For ablation studies, we conducted experiments with DM-Count [10] on NWPU dataset. The loss function involved in optimization (Sec. 3.4) is of the form

$$\mathcal{L}^* = \mathcal{L} + \lambda_2 \widehat{\mathcal{L}} \tag{10}$$

where $\mathcal{L}^*$ is the final loss function, $\mathcal{L}$ is the model loss, $\widehat{\mathcal{L}}$ is the Bin Loss and λ_2 is a weighting factor. From Eqn. 7, we need to tune for both λ_1, λ_2. We conduct a grid optimization with λ_2 ranging over $\{0.01, 1\}$ and λ_1 over $\{1, 10, 100\}$. The pooled MAE and standard deviations are summarized in Table 2. Based on the results, we fix $\lambda_1 = 1, \lambda_2 = 1$ for DM-Count [10] on all datasets and minibatching schemes (RR,RS).

The effectiveness of bin-loss (Eqn. 7) also depends on the extent to which a reference architecture utilizes the formulation for better optimization. For SCAR [2] and SFA-Net [15], we hypothesize that this ability is relatively lower. Therefore, bin-loss is not always better for these networks (see Table 1). Other networks (BL [7], DM-Count [10]) utilize the loss better, leading to consistent improvement in MAE and standard deviation. However, SCAR [2] is still better than no-binning in all cases except NWPU dataset. SFA-Net's performance with bin-loss included is better for the larger UCF, NWPU datasets. Also, inclusion of bin-loss results in consistent gains in terms of error standard deviation especially on the larger, heavily skewed datasets.

As mentioned in Sec. 3.2.1, 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:

$$\begin{aligned} lik(B_k) &= lik(x_1, \ldots, x_{m_k}; \lambda_1, \ldots, \lambda_{m_k}) \\ &= \prod_{j=1}^{m_k} \frac{\lambda_j e^{-\lambda_j}}{x_j} \end{aligned} \tag{11}$$

where $\lambda_1, \ldots, \lambda_{m_k}$ are the parameters of the Poission distributions associated with the bin elements. The other terms are used in the same context as Eqn. 4 in Section 3.2.2. The second candidate distribution for binning is a variant of the multinomial, called stratified multinomial [1]. 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 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 1: Evaluation results on four benchmark datasets NWPU, UCF-QNRF, ShanghaiTech-A,B (STA,STB) using the evaluation procedure in Sec. 3.5 on diverse models. The size of test set is indicated below dataset name. The columns represent minibatching schemes (Bin Loss: random bin selection (RS), 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 blue. The best MAE and standard deviation of the absolute errors are highlighted in bold for each network. Note that gray highlighted columns of the table (Pooled MAE and standard deviation) are not directly comparable to the Global MAE and standard deviation values.

			Pooled MAE and std			Global MAE and std
Size of Dataset	Dataset	Model	Bin loss	Bin loss (RR)	No-binning	Pretrained,No-binning
Large	NWPU 500	DM-Count [10]	$88.1^{\pm 236.7}$	$\mathbf{76.7^{\pm 205.0}}$	$77.8^{\pm 214.9}$	$71.7^{\pm 376.4}$ [10]
		BL [7]	$112.9^{\pm 333.7}$	$114.8^{\pm \mathbf{320.3}}$	$\mathbf{102.5}^{\pm 348.2}$	$102.5^{\pm 560.6}$
		S-DCNet [12]	$213.4^{\pm 231.0}$	$224.1^{\pm \mathbf{230.1}}$	$\mathbf{210.0}^{\pm 303.1}$	$248.7^{\pm 1161.9}$
		SCAR [2]	$112.8^{\pm \mathbf{321.3}}$	$111.9^{\pm 325.6}$	$\mathbf{111.3}^{\pm 332.1}$	$111.3^{\pm 555.8}$
		SFA-Net [15]	$136.0^{\pm 299.1}$	$\mathbf{116.4^{\pm 285.2}}$	$125.0^{\pm 343.0}$	$163.4^{\pm 1072.1}$
	UCF 334	DM-Count [10]	$103.8^{\pm \mathbf{107.5}}$	$97.9^{\pm 109.1}$	$\mathbf{94.5}^{\pm 111.6}$	$85.9^{\pm 120.6}$ [10]
		BL [7]	$\mathbf{91.1^{\pm 100.3}}$	$92.1^{\pm 105.8}$	$98.3^{\pm 134.2}$	$87.1^{\pm 126.8}$ [7]
		S-DCNet [12]	$205.9^{\pm \mathbf{157.8}}$	$\mathbf{199.2}^{\pm 164.8}$	$215.2^{\pm 190.0}$	$214.7^{\pm 277.7}$
		SCAR [2]	$124.5^{\pm \mathbf{128.6}}$	$\mathbf{122.9}^{\pm 129.0}$	$123.4^{\pm 146.9}$	$123.4^{\pm 197.1}$
		SFA-Net [15]	$\mathbf{128.6^{\pm 133.4}}$	$128.9^{\pm 162.9}$	$128.7^{\pm 163.2}$	$128.7^{\pm 199.9}$
Medium	STA 182	DM-Count [10]	$\mathbf{88.6^{\pm 64.4}}$	$89.6^{\pm 75.9}$	$93.0^{\pm 81.3}$	$64.1^{\pm 78.4}$ [10]
		BL [7]	$\mathbf{68.6}^{\pm 69.9}$	$68.9^{\pm 63.3}$	$68.7^{\pm \mathbf{61.9}}$	$63.5^{\pm 74.7}$ [7]
		S-DCNet [12]	$66.6^{\pm 72.6}$	$\mathbf{60.5^{\pm 65.5}}$	$61.3^{\pm 66.9}$	$61.3^{\pm 88.7}$
		SCAR [2]	$83.7^{\pm 67.4}$	$\mathbf{72.9^{\pm 61.8}}$	$79.3^{\pm 67.4}$	$79.3^{\pm 82.9}$
		SFA-Net [15]	$68.4^{\pm 65.1}$	$64.9^{\pm 59.5}$	$\mathbf{63.6^{\pm 55.6}}$	$63.6^{\pm 92.9}$
	STB 316	DM-Count [10]	$9.1^{\pm 9.3}$	$\mathbf{8.6^{\pm 8.6}}$	$8.9^{\pm 10.3}$	$7.3^{\pm 9.3}$ [10]
		BL [7]	$\mathbf{9.6}^{\pm 9.3}$	$9.7^{\pm 9.3}$	$10.8^{\pm \mathbf{9.2}}$	$7.5^{\pm 9.4}$ [7]
		S-DCNet [12]	$9.2^{\pm 9.4}$	$9.6^{\pm 10.5}$	$\mathbf{7.9^{\pm 8.6}}$	$7.8^{\pm 11.0}$
		SCAR [2]	$\mathbf{9.8^{\pm 10.2}}$	$13.8^{\pm 11.7}$	$10.3^{\pm 14.0}$	$10.3^{\pm 14.1}$
		SFA-Net [15]	$9.0^{\pm 7.3}$	$8.8^{\pm 8.0}$	$\mathbf{7.4^{\pm 6.8}}$	$7.4^{\pm 9.2}$

Table 2: Hyperparameter search for λ_1 and λ_2 over a grid and the resulting pooled MAE and standard deviations.

$\lambda_1 \downarrow \lambda_2 \rightarrow$	0.01	1
1	$84.1^{\pm 183.2}$	$76.7^{\pm 205.0}$
10	$80.5^{\pm 243.7}$	$79.5^{\pm 238.7}$
100	$80.7^{\pm 236.8}$	$80.4^{\pm 252.7}$

Table 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	$84.8^{\pm 441.2}$	$89.1^{\pm 533.1}$	$77.8^{\pm 380.3}$
Stratified Multinomial	$90.0^{\pm 283.5}$	$90.6^{\pm 374.0}$	$80.7^{\pm 290.7}$
Multinomial (ours)	$88.1^{\pm 236.7}$	$76.7^{\pm 205.0}$	$77.8^{\pm 214.9}$

6 CONCLUSION

In this paper, we highlight biases at various stages of the typical crowd counting pipeline and propose novel modifications to address issues at each stage. We propose a novel Bayesian sample stratification approach to enable balanced minibatch sampling. Complementary to our sampling approach, we propose a novel loss function to encourage strata-aware optimization. We analyze the performance of crowd counting approaches across standard datasets and demonstrate that our proposed modifications reduce error standard deviation in a noticeable manner. Altogether, our contributions represent a nuanced, statistically balanced and fine-grained characterization of performance for crowd counting approaches.

The proposed bin-aware loss visibly reduces standard deviation of error. However, our work highlights the need for approaches in which error deviations are negligible compared to the mean error. We hope that our work motivates the community to join us in exploring these challenging aspects of the problem. Studying and addressing issues we have raised would enable statistically reliable crowd counting approaches in future.

REFERENCES

[1] Jan Florjanczyk and Taylor Sather. 2015. Stratified Bayesian Blocks.

[2] Junyu Gao, Qi Wang, and Yuan Yuan. 2019. SCAR: Spatial-/channel-wise attention regression networks for crowd counting. *Neurocomputing* 363 (2019), 1–8.

[3] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. 2018. Composition Loss for Counting, Density Map Estimation and Localization in Dense Crowds. In *ECCV*.

[4] Daniel Jurafsky and James H. Martin. 2000. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (1st ed.). Prentice Hall PTR, USA.

[5] Yuhong Li, Xiaofan Zhang, and Deming Chen. 2018. CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes. *CoRR* abs/1802.10062 (2018).

[6] W. Liu, M. Salzmann, and P. Fua. 2019. Context-Aware Crowd Counting. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

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

[8] D. B. Sam, S. Surya, and R. V. Babu. 2017. Switching Convolutional Neural Network for Crowd Counting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 4031–4039.

[9] Jeffrey D. Scargle, Jay P. Norris, Brad Jackson, and James Chiang. 2013. STUDIES IN ASTRONOMICAL TIME SERIES ANALYSIS. VI. BAYESIAN BLOCK REPRESENTATIONS. *The Astrophysical Journal* 764, 2 (Feb 2013), 167.

[10] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. 2020. Distribution Matching for Crowd Counting. In *Advances in Neural Information Processing Systems*.

[11] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. 2020. NWPU-Crowd: A Large-Scale Benchmark for Crowd Counting and Localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[12] Haipeng Xiong, Hao Lu, Chengxin Liu, Liu Liang, Zhiguo Cao, and Chunhua Shen. 2019. 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)*. 8362–8371.

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

[14] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 589–597.

[15] Liang Zhu, Zhijian Zhao, Chao Lu, Yining Lin, Yao Peng, and Tangren Yao. 2019. Dual Path Multi-Scale Fusion Networks with Attention for Crowd Counting. (2019).