Uniformity in Heterogeneity: 
Diving Deep into Count Interval Partition for Crowd Counting

Changan Wang1* Qingyu Song1* Boshen Zhang1 Yabiao Wang1
Ying Tai1 Xuyi Hu1,3 Chengjie Wang1† Jilin Li1 Jiayi Ma4 Yang Wu2
1Tencent Youyu Lab, 2Applied Research Center (ARC), Tencent PCG
3Department of Electronic & Electrical Engineering, University College London, United Kingdom
4Electronic Information School, Wuhan University, Wuhan, China
{changanwang, boshenzhang, casexhu, yingtai, jasoncjwang, jerolinli}@tencent.com,
qingyusong@zju.edu.cn, zceexhu@ucl.ac.uk, jyma2010@gmail.com, dylanywu@tencent.com

Abstract

Recently, the problem of inaccurate learning targets in crowd counting draws increasing attention. Inspired by a few pioneering work, we solve this problem by trying to predict the indices of pre-defined interval bins of counts instead of the count values themselves. However, an inappropriate interval setting might make the count error contributions from different intervals extremely imbalanced, leading to inferior counting performance. Therefore, we propose a novel count interval partition criterion called Uniform Error Partition (UEP), which always keeps the expected counting error contributions equal for all intervals to minimize the prediction risk. Then to mitigate the inevitably introduced discretization errors in the count quantization process, we propose another criterion called Mean Count Proxies (MCP). The MCP criterion selects the best count proxy for each interval to represent its count value during inference, making the overall expected discretization error of an image nearly negligible. As far as we are aware, this work is the first to delve into such a classification task and ends up with a promising solution for count interval partition. Following the above two theoretically demonstrated criterions, we propose a simple yet effective model termed Uniform Error Partition Network (UEPNet), which achieves state-of-the-art performance on several challenging datasets. The codes will be available at: TencentYouyuResearch/CrowdCounting-UEPNet.

1. Introduction

The task of crowd counting estimates the number of people in given images or videos. It has drawn remarkable attention in recent years, since its wide range of practical applications in public safety. Currently, most existing state-of-the-art methods adopt density maps [1, 18, 19] as the targets and learn with Convolutional Neural Networks (CNNs).

However, the widely used learning targets, i.e., the density maps, are actually inaccurate and tend to be noisy, as discussed in [1, 12, 17, 20, 23]. These imperfect density maps are caused by several factors, such as empirically selected Gaussian kernels, large density variations and label-
ing deviations. As shown in the supplementary materials, the inaccurate density map introduces severe inconsistency between the semantic content and the ground truth target, acting as a kind of “noises” in the learning target. Therefore, when learning the exact count values with an outlier-sensitive loss, such as the commonly used Mean Squared Error (MSE), the model may tend to overfit these inaccurate and ambiguous “noises”, leading to inferior performance and poor generalization ability [12].

Inspired by a few pioneering works [3, 12, 21, 25], we solve the above problem with the paradigm of local count (the count value of a local patch) classification, as shown in Figure 1. In such a paradigm, the local count is quantified to one of several pre-defined count interval bins, depending on the interval it falls into. Then we resort to count interval classification by taking each interval as a single class. However, the number of samples with increasing local counts follows a long-tailed distribution, leading to imbalanced training data and extreme counting errors for some count intervals. What makes the problem worsen is that samples from different intervals have various learning difficulty, thus also contribute different counting errors. In this paper, firstly, we solve the above problem from the perspective of designing an optimal interval partition strategy. Specifically, we derive an Uniform Error Partition (UEP) criterion following the principle of maximum entropy. Without any prior over the density distribution of an unseen image, the UEP criterion keeps the expected counting errors nearly equal for all intervals to minimize the prediction risk. Such a novel strategy provides a comprehensive consideration of misclassification errors and sample imbalance problem.

Secondly, another crucial yet unsolved problem of this paradigm raises from inference stage. Specifically, a fixed value (i.e., count proxy) should be selected from each count interval, using as the predicted counts for those patches classified to this interval. Obviously, mapping a range of local counts to a single count value would inevitably introduce perceivable discretization errors. To mitigate the above problem, we propose a Mean Count Proxies (MCP) criterion for the count proxy selection. The MCP criterion uses average count value of samples in an interval as the optimal count proxy, with which the expected discretization error is theoretically demonstrated to be nearly negligible.

Thirdly, since the distribution of count values is essentially continuous, it is tricky to unambiguously classify samples whose count values are located around the interval borders. Thus, we design two parallel prediction heads with overlapping count intervals, named Interleaved Prediction Heads (IPH). In this reciprocal prediction structure, the samples whose count values fall near the interval borders of one head are more likely to be correctly classified in another head, reducing the misclassification errors due to the learning ambiguity around interval borders. We conduct extensive experiments on several public datasets with various crowd densities to show the consistent improvements. The contributions of this paper are summarized as follows:

1. We propose a novel criterion termed UEP to tackle with the crucial yet unsolved count interval partition problem, making the local count classification based counting models achieve the minimum prediction risk.

2. We propose the MCP criterion to minimize the discretization errors inevitably introduced by the count quantification, which is orthogonal to the UEP criterion.

3. We propose a simple yet effective model termed UEPNet, following the above two key criteria. Combining with the IPH, the UEPNet achieves state-of-the-art performance on several benchmarks.

2. Related Work

In this section, we review the recent density map learning based methods in the literature. Especially, we reveal the defects in very few existing local count classification based methods to show the necessity of our research. We also discuss some works dealing with the imperfect targets to show the superiority of our method.

**Pixel-wise Density Maps Learning.** The concept of density maps is firstly introduced by Lempitsky and Zisserman [10], and since then this kind of learning targets has been widely used in subsequent methods [2, 8, 11, 28]. These methods are trained with pixel-wise supervision and finally obtain the estimated count by summing over the predicted density maps. Although these methods [5, 9] achieved good performance, they still suffer from the following defects. Firstly, they ignore the negative impacts of the imperfect density maps and learn to overfit these inaccurate targets with an outlier-sensitive loss. Secondly, there exists inconsistency between the minimum training loss and the optimal counting result. These problems limit the counting accuracy and generalization ability of such approaches.

**Patch-wise Density Maps Learning.** Instead of pixel-wise regression, patch-wise density maps learning based methods predict the count values of local patches. This idea is firstly used in S-DCNet [25], and is helpful to alleviate the inconsistency between training target and evaluation criteria as demonstrated in [16]. Among these methods, some works [24] adopt regression with supervision from the commonly used MSE loss but still suffer from the inaccurate targets. Thus the other methods [12, 25] discretize the local counts and learn to classify count intervals. However, despite the robustness of such a paradigm, both of them divide the count intervals intuitively, which is inappropriate and greatly limits the performance. Besides, the two methods directly use median value of the interval to represent its class count, which is sub-optimal and introduces additional discretization errors. We also adopt the paradigm of patch-
Learning with Inaccurate Targets. Recently, there has been increasing attention on the problem of inaccurate ground truth targets. One solution is to improve the quality of the generated density maps by estimating more accurate Gaussian kernel [20, 23], while the improvement seems to be limited by insufficient annotations. Bayesian loss is proposed by [17] to provide a more reliable supervision on the counting difference from multiple images or patches, in which the counting difference is calculated by the subtraction of ground truth count and predicted count.

3. Our Approach

We firstly introduce the symbols used in this paper in Sec. 3.1. After that, we introduce the pipeline of our framework in detail. Specifically, we begin with the calculation of the local counts in an image as described in Sec. 3.2. Then we exhibit our strategy of quantifying these local counts during the online updating. Differently, we resort to a pixel-wise regression. Moreover, ADSCNet [1] proposes a self-correction supervision to iteratively correct the annotations with the model estimations, which however ignores the additional noises introduced by those inaccurate estimations during the online updating. Differently, we resort to local count classification, which is more robust than learning the exact but inaccurate count values [12].

3.1. Symbol Definition

Table 1 lists the main symbols used in the following text. In this paper, we conduct a patch-level analysis by breaking down the whole training set into $K$ image patches with a same size. For the $k$-th image patch, the summation of its density values is calculated as its local count $d_k$. Then there should be a collection $\mathcal{T}$ containing all the local counts of the above $K$ samples. For an arbitrary partition setting with $m$ intervals, we denote the $i$-th interval as $c_i$. And the length $l_i$ of $c_i$ is determined by its left border $t_i$ and right border $t_{i+1}$. In other words, the count interval $c_i$ is a contiguous count value range from $t_i$ to $t_{i+1}$. Then for any sample $x_{ij}$ ($j \in \{1, ..., n_i\}$) whose local count falls into the interval $c_i$, we assign a fixed count value $\delta_i$ (termed as count proxy) as its predicted count if it is correctly classified. $\mathcal{E}$ is the expectation of the counting error for a randomly given unseen image. $\mathcal{E}$ is the absolute value of the average counting difference from multiple images or patches, in which the counting difference is calculated by the subtraction of ground truth count and predicted count.

3.2. Local Count Calculation

For an image containing $n$ heads, the ground truth annotations can be expressed as a dot map $M = \sum_{i=1}^{n} \delta(p - p_i)$, where $p_i$ is the point annotation for the $i$-th head and the delta function $\delta(p - p_i)$ represents there is one head at pixel $p_i$. Then we generate density map $D$ by convolving over $M$ using a Gaussian kernel $G_{\sigma}$, which can be described as
\[ D = \sum_{i=1}^{n} \delta(p - p_i) \ast G_{\sigma}. \] The spread parameter \( \sigma \) in \( G_{\sigma} \) is fixed or determined by the \( k \) nearest neighbors \([28]\). With a normalized Gaussian kernel, the counting results can be obtained by summing over the density maps.

Based on the density map \( D \), we can construct a local count map \( D_s \), in which each value is the head count from the corresponding patch of size \( s \times s \) in the image. Specifically, we slide a window of size \( s \times s \) without overlapping over the density map \( D \) and sum the density values in this window as the local count \( d \).

### 3.3. The Interval Partition Strategy

We derive our key partition strategy from the perspective of minimizing the expected counting error \( \tilde{E} \) for an unseen image. And \( E \) is obtained by calculating the absolute value of the difference between the estimated count and the ground truth count at the image level. Since we have no prior over the density distribution of any given unseen image, we propose to approximate \( E \) from a perspective of probability sampling. Specifically, we firstly decompose \( E \) as the absolute value of the summation for the counting differences from multiple non-overlapping patches of that image. Then under the assumption of independent and identically distributed (i.i.d.) data, these images patches are seen as a random sampling subset from \( K \) image patches in the training set. Thus, minimizing \( E \) for a randomly given image is equivalent to achieving the minimum \( \tilde{E} \) on all patches in the sampled subset. The above approximation helps us to design a generalized partition strategy using the available training set, as demonstrated in Supplementary Materials.

Since our goal is to minimize \( \tilde{E} \) with a proper count interval partition strategy, we would like to figure out how the counting errors from different intervals contribute to \( \tilde{E} \). However, for any given unseen image, the patches in its equivalent subset are always randomly selected. In other words, for a specific image, we have no way to determine the patch distribution over all intervals, neither its counting error contribution from different intervals. In such a case, according to the principle of maximum entropy, we should not make further assumptions to this distribution. As a result, we propose an Uniform Error Partition (UEP) criterion, which always keeps the expected counting error equal for all intervals. With the proposed UEP criterion, the risk of making extreme prediction error is minimized, achieving reasonable \( \tilde{E} \) for any given image. From another perspective, we could achieve a minimum \( \tilde{E} \) in terms of mathematical expectation with the UEP criterion.

Before conducting the above UEP criterion, a key step is to estimate the counting error from a certain interval. In this part, we estimate the interval-level count error based on all samples in \( \mathcal{T} \), because every element within it has the same chance to be selected into the equivalent subset for an image. For a specific interval \( c_i \), its counting error contribution depends on both the sample number \( n_i \) within it and the misclassification cost of each sample from it. The former one is proportional to the interval length \( l_i \), which is relatively easier to obtain from the dataset statistics. Besides, for a single sample \( x_{ij} \) from interval \( c_i \), it is more likely to be misclassified to a nearby interval. Since the interval lengths of adjacent intervals are nearly equal, the misclassification cost is also proportional to the interval length \( l_i \). Therefore, we reasonably take \( n_i l_i \) as the approximation for the counting error from \( c_i \). By keeping a uniform \( n_i l_i \) for all intervals, the UEP criterion takes a fully consideration on the misclassification cost and the sample imbalance problem among intervals, making the classification task more easier to learn. We provide more detailed analysis in the Supplementary Materials.

To apply the UEP criterion during the interval partition process, we propose a Binary Interval Partition (BIP) algorithm, as shown in Algorithm 1. For a required interval number \( m \), the BIP algorithm determines a target \( \bar{n} \bar{l} \) for all intervals by an iterative binary searching process. By

---

**Algorithm 1: Binary Interval Partition with UEP**

**Input:** \( \mathcal{T} \leftarrow \) local count collection in the training set

\[ m \leftarrow \text{required number of intervals} \]

\[ [L, H] \leftarrow \text{search range for} \ n_i l_i \]

**Output:** \( \mathcal{P} \leftarrow \) interval endpoint collection

\[ T \leftarrow \text{sort}(\text{filter}(\mathcal{T})) \]

while \( \text{abs}(H - L) > \epsilon \) do

\[ n \leftarrow n + 1 \]

/* try to divide by \( n_i l_i = \bar{l} \) */

if \( (d_k - p) \times n > \bar{l} \) then

\[ \begin{align*}
    n &\leftarrow 0; \ n \leftarrow 0; \mathcal{P} \leftarrow \{t_0\}, \bar{l} \leftarrow (L + H)/2 \\
    &\text{/* adjust } \ [L, H] \text{ by } |\mathcal{P}| \text{ and } m \}*
\end{align*} \]

if \( |\mathcal{P}| = m \) then

\[ \begin{align*}
    L &\leftarrow \bar{l} \quad \text{// when } |\mathcal{P}| = m \\
    &\text{/* adjust by } n_{m-1} l_{m-1} \}*
\end{align*} \]

if \( (t_{\text{MAX}} - p) \times n > \bar{l} \) then

\[ \begin{align*}
    L &\leftarrow \bar{l} \\\n    &\text{// when } |\mathcal{P}| < m - 1
\end{align*} \]

return \( \text{sort}(\mathcal{P}) \)
difficult to be correctly classified, which is due to the learning ambiguity around the interval borders. These samples tend to contribute more to the misclassification errors, especially for those samples from the intervals with relatively larger length.

To alleviate the above problem, we parallelly duplicate the prediction head and use overlapping count intervals for them. Specifically, as shown in Figure 3, the interval borders in the first head (PredHead0) are obtained using the UEP criterion. While the other head (PredHead1) directly use the count proxies in PredHead0 as its interval borders (except for the count interval \([0, t_0)\) of the background class). The count proxies of PredHead1 are independently calculated using the MCP criterion. The two heads are trained with CE loss independently, and during inference their predicted local count maps are averaged as the final estimation. With the IPH, the samples whose count values fall near the interval borders of one head are more likely to be correctly classified in the other head.

### 4. Experiments

We conduct extensive experiments on four challenging benchmarks with various crowd densities. Mean Absolute Error (MAE) and Root Mean Squared Error (MSE) are used as the evaluation metrics following [28].

#### 4.1. Implementation Details

**Datasets.** Experimental evaluations for the UEPNet are conducted on four widely used crowd counting benchmark datasets: ShanghaiTech [28] part A and part B, UCF-CC_50 [6], UCF-QNRF [7], and WorldExpo’10 [27]. The count statistics for these datasets can be found in a recent survey [4]. We generate ground truth density maps for ShanghaiTech part B and WorldExpo’10 using Gaussian kernel \(G_{\sigma}\) with fixed sigma of 15.0 and 5.0 respectively. For the other datasets we use geometry-adaptive Gaussian kernel following [28]. We follow the standard protocol in [6] to conduct a five-fold cross validation on UCF-CC_50. For WorldExpo’10, we blur image regions outside the provided RoIs using mean filter of kernel size 11.

**Data Augmentations.** Scaling with a random factor in \([0.3, 1.3]\) and random mirroring are performed to increase the diversity of image patches. Then we random select a cropped patch for training following CSRNet [11]. Since images in UCF-QNRF have extremely large resolution, we limit the crop size within 1024 and make inference with a sliding window of size 1024 due to the limited memory.

**Hyperparameters.** The size of local patches in UEPNet is \(8 \times 8\), since we use the feature map of stride 8 for predictions. As demonstrated, the interval number works well in a wide range, and is empirically selected as 25 for all datasets. The \(c\) in Algorithm 1 is set to 100, and \(t_0\) is set to \(1.6e^{-4}\).
The first 13 convolutional layers in UEPNet are initialized from a pre-trained VGG-16 [22] model, and the rest layers are randomly initialized by a Gaussian distribution with the mean of 0 and the standard deviation of 0.01. We fine-tune the model using SGD with batch size 1. The learning rate is initially set to 0.001 and is decreased by 1/10 when the training loss stagnates. We keep training until convergence.

4.2. Comparisons with State-of-the-Art Methods

We compare our UEPNet with state-of-the-art methods on the datasets described above. As shown in Table 2, our UEPNet achieves state-of-the-art performance on these datasets. Compared with the previous strongest method ADSCNet, which achieved the best MAE on several datasets, UEPNet achieves better MAE on SHTech Part B. And the UEPNet surpasses ADSCNet with a 1.4% relative improvement in MAE on SHTech Part B, with a 16.7% relative improvement in MAE on UCF_CC_50. As for UCF-QNRF, most methods give very few details of the testing procedure including ADSCNet. With a simple patch-wise inference (due to the limited GPU memory), our UEPNet achieves higher MAE than ADSCNet, but already reports significant improvements indicate the crucial of count interval design and the effectiveness of our UEP criterion.

4.3. Ablation Studies

We further make comparison with the other two methods, Two-Linear in S-DCNet [25] and Liu et al. [12], to show the superiority of our strategy for interval partition and count proxy selection. It is worth mentioning that the two methods also use the paradigm of counting by local count classification, but both of them divide the count intervals intuitively. As shown in Table 2, our UEPNet achieves significant improvement on all the four datasets. For more fair comparison, we firstly remove the IPH and use median value of the interval as its count proxy. Then in order to use their carefully tuned hyperparameters, we conduct experiments under different strides with the same network structure and report the results in Table 3. The consistent and significant improvements indicate the crucial of count interval design and the effectiveness of our UEP criterion.

### Table 2: Comparisons with SOTA methods on four datasets. The best performance is bold and the second best is underlined.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SHTechPartA</th>
<th>SHTechPartB</th>
<th>UCF_CC_50</th>
<th>UCF-QNRF</th>
<th>WorldExpo’10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>MSE</td>
<td>MAE</td>
<td>MSE</td>
<td>S1</td>
</tr>
<tr>
<td>CAN [15]</td>
<td>62.3</td>
<td>100.0</td>
<td>7.8</td>
<td>12.2</td>
<td>212.2</td>
</tr>
<tr>
<td>CSNet [11]</td>
<td>68.2</td>
<td>115.0</td>
<td>10.6</td>
<td>16.0</td>
<td>266.1</td>
</tr>
<tr>
<td>PGCNet [20]</td>
<td>57.0</td>
<td>86.0</td>
<td>8.8</td>
<td>13.7</td>
<td>244.6</td>
</tr>
<tr>
<td>DSSNet [14]</td>
<td>60.63</td>
<td>96.04</td>
<td>6.85</td>
<td>10.34</td>
<td>216.9</td>
</tr>
<tr>
<td>Liu et al. [12]</td>
<td>62.8</td>
<td>102.0</td>
<td>8.6</td>
<td>16.4</td>
<td>239.6</td>
</tr>
<tr>
<td>BL+ [17]</td>
<td>62.8</td>
<td>101.8</td>
<td>7.7</td>
<td>12.7</td>
<td>229.3</td>
</tr>
<tr>
<td>S-DCNet [25]</td>
<td>58.3</td>
<td>95.0</td>
<td>6.7</td>
<td>10.7</td>
<td>204.2</td>
</tr>
<tr>
<td>DUBNet [19]</td>
<td>64.6</td>
<td>106.8</td>
<td>7.7</td>
<td>12.5</td>
<td>243.8</td>
</tr>
<tr>
<td>SDAnet [18]</td>
<td>63.6</td>
<td>101.8</td>
<td>7.8</td>
<td>10.2</td>
<td>227.6</td>
</tr>
<tr>
<td>ADSCNet [1]</td>
<td>55.4</td>
<td>97.7</td>
<td>6.4</td>
<td>11.3</td>
<td>198.4</td>
</tr>
<tr>
<td>LibraNet [13]</td>
<td>55.9</td>
<td>97.1</td>
<td>7.5</td>
<td>11.3</td>
<td>181.2</td>
</tr>
<tr>
<td>AMRNet [5]</td>
<td>56.7</td>
<td>93.4</td>
<td>6.7</td>
<td>10.2</td>
<td>208.4</td>
</tr>
<tr>
<td>ASNet [9]</td>
<td>57.78</td>
<td>90.13</td>
<td>-</td>
<td>-</td>
<td>174.84</td>
</tr>
<tr>
<td>AMRNet [16]</td>
<td>61.59</td>
<td>98.36</td>
<td>7.02</td>
<td>11.00</td>
<td>184.0</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>54.64</strong></td>
<td><strong>91.15</strong></td>
<td><strong>6.38</strong></td>
<td><strong>10.88</strong></td>
<td><strong>165.24</strong></td>
</tr>
</tbody>
</table>

### Table 3: Comparisons with other interval partition methods on SHTech Part B using the same network structure and the same number of intervals.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Stride8</th>
<th>Stride16</th>
<th>Stride32</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAE</td>
<td>7.72</td>
<td>9.63</td>
<td>7.71</td>
</tr>
<tr>
<td>MSE</td>
<td>12.51</td>
<td>14.22</td>
<td>12.80</td>
</tr>
</tbody>
</table>

### Table 4: Ablation studies of our contributions. MS in the table is short for the multi-scale training. PPH is a variant of the IPH, and the two heads in the PPH use exactly the same interval setting instead of the overlapping intervals.

<table>
<thead>
<tr>
<th>MCP</th>
<th>IPH</th>
<th>PPH</th>
<th>MS</th>
<th>SHTechPartA</th>
<th>SHTechPartB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61.56</td>
<td>102.30</td>
<td>7.72</td>
<td>12.51</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>58.69</td>
<td>97.64</td>
<td>7.64</td>
<td>13.18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>57.97</td>
<td>100.34</td>
<td>6.87</td>
<td>11.33</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>56.80</td>
<td>92.01</td>
<td>7.25</td>
<td>11.91</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>58.47</td>
<td>97.51</td>
<td>7.57</td>
<td>12.88</td>
<td></td>
<td></td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>54.64</strong></td>
<td><strong>91.15</strong></td>
<td><strong>6.38</strong></td>
<td><strong>10.88</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

In this section, we demonstrate the effectiveness of our contributions through ablation studies on two datasets with various density, i.e., SHTech Part A and SHTech Part B.

**Effect of UEP.** As shown in Table 3, dividing count intervals with our UEP criterion performs better than the previous two intuitively designed interval settings, with 20.0%
<table>
<thead>
<tr>
<th>Methods</th>
<th>Median</th>
<th></th>
<th>MCP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>MSE</td>
<td>MAE</td>
</tr>
<tr>
<td>UniformLen</td>
<td>82.62</td>
<td>90.65</td>
<td>13.14</td>
</tr>
<tr>
<td>UniformNum</td>
<td>62.46</td>
<td>75.26</td>
<td>10.88</td>
</tr>
<tr>
<td>Ours</td>
<td>7.72</td>
<td>12.51</td>
<td>7.64</td>
</tr>
</tbody>
</table>

Table 5: Comparison with other interval settings on SHTech Part B. We use Median to represent the median value of each interval as its count proxy.

and 13.2% improvements on MAE respectively. To highlight the benefit of UEP criterion, which takes a comprehensive consideration of the misclassification cost and the sample imbalance problem among intervals, we make comparison with another two interval settings, i.e., UniformLen and UniformNum. UniformLen represents the strategy of dividing intervals with equal interval length, and UniformNum represents the strategy of dividing intervals with equal number of samples in each interval. From Table 5, our method significantly outperforms UniformLen and UniformNum.

**Effect of MCP.** As shown in Table 5, compared with the median count proxies, the adoption of our MCP consistently reduces the MAE of UniformLen and UniformNum, with 20.0% and 13.2% reduction respectively. From the ablation studies in Table 4, with the MCP, the MAE improves from 61.56 to 58.69 on Part A and from 7.72 to 7.64 on Part B.

**Effect of IPH.** From the ablation results in Table 4, the IPH indeed reduces the estimation errors, with 3.2% and 5.1% improvements on MAE for SHTech Part A and SHTech Part B respectively. Due to the augmentation of multi-scale training, the relative improvements for IPH are a little higher (5.7% and 7.1%). To eliminate the gain of voting ensemble, we conduct experiments using Parallel Prediction Heads (PPH), which uses exactly the same interval setting instead of the overlapping intervals for the two heads. As shown in Table 4, the improvement of PPH is only 0.4% for SHTech Part A and 0.9% for SHTech Part B.

**Effect of Multi-Scale Training.** This data augmentation is particularly helpful to the classification related task, i.e., the local count classification, which significantly increases the diversity of samples in each interval. As shown in Table 4, with multi-scale training, the MAE of our UEPNet reduces from 56.80 to 54.64 on SHTech Part A and from 7.25 to 6.38 on SHTech Part B. However, as demonstrated in the following, the benefit of multi-scale training is only valid for the paradigm of counting by local count classification.

### 4.4. Discussions

**Why classification works better?** It is natural to ask that why the task of local count classification could perform better than regression task. Besides the promising counting accuracy as shown in [12, 13, 25], we provide an intuitive explanation. Firstly, it only adopts a Cross Entropy (CE) loss, which is more robust to outliers compared with the commonly used MSE loss in regression. Secondly, it is much easier to identify that if a local count falls into a certain interval than determining its accurate value. Thirdly, the regression task requires to strictly predict the exact counts of two patches, although their ground truth counts might be inaccurate. Differently, for the classification task, only when the gap between their counts is large enough (beyond the length of an interval), the two patches might be classified as different classes. Under the existence of inaccurate targets, this more relaxed supervision helps to stabilize the training process, avoiding overfitting or local optimum. Finally, such a paradigm shows its potential of achieving superior performance when using our proposed criterions.

---

**Figure 4: The effect of the number of intervals.**

**How to determine the class number?** We conduct experiments with different number of intervals. As shown in Figure 4, our UEPNet is robust to the number of intervals, while the methods of UniformLen and UniformNum get much higher MAE when using lower number of intervals. As we increase the number of intervals, the MAE of UniformLen and UniformNum starts to reduce and finally stagnates, but is still higher than our method. Consequently, what matters is the interval partition strategy rather than the number of intervals, and our method is insensitive to the number of intervals. We empirically recommend to use 25 classes, which works well on all dataset in our experiments.

<table>
<thead>
<tr>
<th></th>
<th>S-DCNet</th>
<th>Liu et al.</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Median</td>
<td>17.84</td>
<td>11.04</td>
<td>1.88</td>
</tr>
<tr>
<td>MCP</td>
<td>3.20</td>
<td>1.70</td>
<td>0.27</td>
</tr>
</tbody>
</table>

Table 6: The discretization errors on SHTech Part B test set.

**The analysis of discretization errors.** We quantify the discretization errors by assuming all samples are correctly classified, which eliminates the impact from the misclassification errors. Specifically, for the local count $d_k$ of each patch (without overlapping) in an image from $I_{test}$, if $d_k$ falls into an interval $I_k$, $I_k \in \{0, 1, \ldots, m - 1\}$, its predicted count $\hat{d}_k$ should be the count proxy of interval $I_k$, i.e., $\hat{d}_k = \delta_{I_k}$. Then, we calculate $| \sum_{d_k \in I_{test}} (d_k - \hat{d}_k) |$ as the discretization errors on $I_{test}$, since no misclassification errors are introduced during this process. As shown in Table 6, with MCP, the average discretization errors of all test
images are significantly reduced. Combining with UEP and MCP, the discretization errors are nearly negligible.

The effectiveness of our interval partition strategy. As illustrated in Figure 5, compared with the commonly used MSE loss, our strategy is less sensitive to the samples with larger prediction error. This advantage is helpful to highlight the useful as well as accurate gradients and mitigate the negative impacts of inaccurate targets. Besides, since we divide intervals with the UEP criterion, which tries to keep the counting error cost equal for all intervals to minimize the prediction risk. We also compare the error contribution of each interval in a converged UEPNet. As shown in Figure 6c, error contributions of all intervals are approximately equal. An interesting observation is that the intervals which are closer to the middle contribute slightly more counting errors. One possible explanation is that the samples in these intervals are more difficult to learn, since they may be either overestimated or underestimated.

<table>
<thead>
<tr>
<th>Methods</th>
<th>without MS</th>
<th>with MS</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MAE</td>
<td>MSE</td>
</tr>
<tr>
<td>MSE</td>
<td>13.57</td>
<td>22.64</td>
</tr>
<tr>
<td>UniformLen</td>
<td>12.71</td>
<td>20.38</td>
</tr>
<tr>
<td>UniformNum</td>
<td>11.25</td>
<td>26.15</td>
</tr>
<tr>
<td>Ours</td>
<td>7.64</td>
<td>13.18</td>
</tr>
</tbody>
</table>

Table 7: The effect of multi-scale training.

Multi-scale training is only useful for classification. Our UEPNet benefits from the data augmentation of multi-scale training. However, we find that multi-scale training is only useful for the paradigm of local count classification. As shown in Table 7, when learn to regress the exact count in each patch with MSE, the adoption of multi-scale training unexpectedly hurts the performance of the model. Although multi-scale augmentation significantly increases the diversity of samples in each interval, but also introduces much more noises due to the inaccurate targets. Consequently, using outlier-sensitive losses will force the model to overfit these noises, which leads to inferior performance. But for local count classification, the benefits of multi-scale augmentation outweigh its side effects. Additionally, compared

![Figure 5: Comparison of gradient contributions in the converged models using MSE and our approach.](image)

![Figure 6: Illustrations of counting error contributions (class MAE) from each count interval for three types of interval partition strategies. The number of samples in each count interval (class) is indicated by different colors in the top jet colormap. (a) UniformLen leads to extremely imbalanced number of samples among intervals, which makes it hard to train a well-performed classifier. (b) UniformNum leads too large interval length for certain intervals (especially for the intervals with high count), thus the misclassification of which contribute too large counting errors. (c) Our method provides a comprehensive consideration of the misclassification cost and the sample imbalance problem.](image)

with UniformLen and UniformNum, we find that the proposed UEP criterion is helpful for maximizing the relative improvement of the multi-scale augmentation.

5. Conclusion

We propose a simple yet effective crowd counting model termed UEPNet based on the paradigm of local count classification. Specifically, we conduct deeply research on this paradigm, and propose the UEP criterion and the MCP criterion. The UEP criterion is used for the interval partition, which divides intervals with equal counting error cost for all intervals to minimize prediction risk. The MCP criterion is used for count proxy selection, which selects the average count value of samples in corresponding interval as its optimal count proxy. Additionally, we also propose the IPH to mitigate the problem of classification ambiguity for samples near the interval borders. Extensive experiments on four widely used datasets with various crowd densities demonstrate the superiority of our approach.
References


[12] Liang Liu, Hao Lu, Haipeng Xiong, Ke Xian, Zhiguo Cao, and Chunhua Shen. Counting objects by blockwise classification. TCSVT, 2019. 1, 2, 3, 6, 7


[17] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In ICCV, 2019. 1, 3, 6


[26] Zhaoyi Yan, Yucheng Yuan, Wangmeng Zuo, Xiao Tan, Yezehe Wang, Shilei Wen, and Errui Ding. Perspective-guided convolution networks for crowd counting. In ICCV, 2019. 6
