This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2021.3056518, IEEE Transactions on Pattern Analysis and Machine Intelligence

Locality-Aware Crowd Counting

Joey Tianyi Zhou, Le Zhang, Jiawei Du, Xi Peng, Zhiwen Fang , Zhe Xiao and Hongyuan Zhu

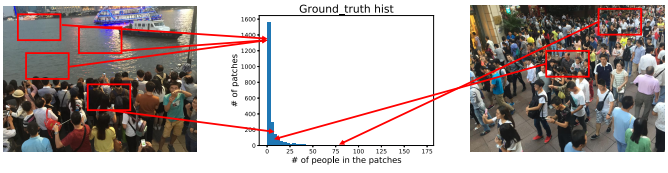
Abstract—Imbalanced data distribution in crowd counting datasets leads to severe under-estimation and over-estimation problems, which has been less investigated in existing works. In this paper, we tackle this challenging problem by proposing a simple but effective locality-based learning paradigm to produce generalizable features by alleviating sample bias. Our proposed method is locality-aware in two aspects. First, we introduce a locality-aware data partition (LADP) approach to group the training data into different bins via locality-sensitive hashing. As a result, a more balanced data batch is then constructed by LADP. To further reduce the training bias and enhance the collaboration with LADP, a new data augmentation method called locality-aware data augmentation (LADA) is proposed where the image patches are adaptively augmented based on the loss. The proposed method is independent of the backbone network architectures, and thus could be smoothly integrated with most existing deep crowd counting approaches in an end-to-end paradigm to boost their performance. We also demonstrate the versatility of the proposed method by applying it for adversarial defense. Extensive experiments verify the superiority of the proposed method over the state of the arts.

Index Terms—Long-tail Distribution, Data-imbalance Learning, Data Augmentation, Crowd Counting, Adversarial Defense.

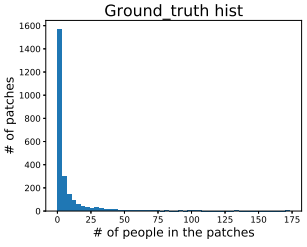
1 Introduction

Crowd counting has attracted increasing attention recently due to its wide-ranging applications in video surveillance, metropolis security, scene understanding, human behavior analysis, and resource management. In past several years, numerous methods have been proposed to estimate the number of people in an image. For example, Lempitshky and Zisserman [1] recast this task from the perspective of automated counting of objects. Similar to other vision tasks, the pioneer crowd counting methods [2], [3] are based on hand-crafted features [4] which are specifically designed by domain experts and may be unable to reveal the latent data distribution. More recently, several learning based methods [5], [6], [7] have shown significant improvements in counting performance, especially in some complex scenarios [8], [9]. Unfortunately, in those complex scenarios, handcrafted features are not powerful enough to handling occlusions, scale variations, high contrast, and so on.

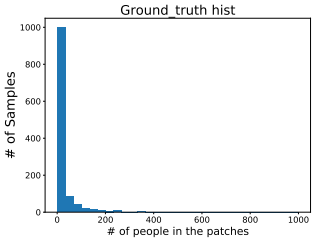
To achieve crowd counting in complex scenarios, recent attention has shifted into deep learning. These works usually employ convolutional neural networks (CNNs) to extract feature in a data-driven way, which could be roughly classified into two categories, i.e., the counts based methods and the density-map based methods. The counts based methods usually learn a regression function to predict people number with the learned deep representations [8], [10], [11], [12]. Although these methods have achieved good performance, most of them ignore the spatial layout, thus weakening their generalization ability to new scenarios. To solve this problem, some density-map based methods are proposed by introducing



(a) Examples of Patches



(b) ShanghaiTech Part B



(c) UCF-QNRF

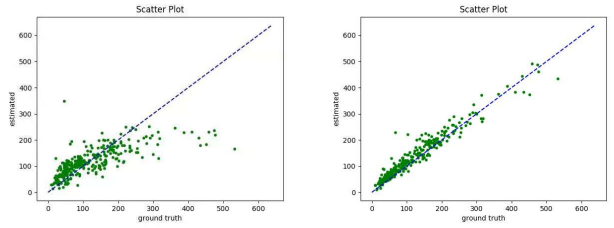
Fig. 1. Highly Imbalanced Training Original Data

salience information for counting and have shown effectiveness in performance improvement. Furthermore, in practice, the crowd images are in a large variance in terms of the crowd density and volume, thus making difficulty in number estimation. To handle these issues especially to improve the model generalization ability, a lot of efforts has been devoted by aggressively exploring deeper model [13] or wider architectures [9] or heuristic engineering approaches [14], [10] or multitask learning pipeline [11]. Among these works, the density-map based approaches [9], [15], [16] with the standard structure of "convolution + pooling" have achieved state-of-the-art performance in the crowd counting task.

Compared with other vision tasks, one major challenge faced by crowd counting is the extremely imbalanced data distribution. Such a data characteristic will lead to inferior counting performance caused by over-estimation or under-estimation [11]. Due to the inherit difficulty, such a data characteristic has less been investigated in existing works. To quantitatively show this case, we present the statistics of two datasets, i.e., ShanghaiTech part

- *Joey Tianyi Zhou, Jiawei Du, Zhe Xiao are with the Institute of High Performance Computing (IHPC), the Agency for Science, Technology and Research (A*STAR), Singapore 138632.*
- *Le Zhang, and Hongyuan Zhu are with the Institute for Infocomm Research (I2R), the Agency for Science, Technology and Research (A*STAR), Singapore 138632.*
- *Xi Peng is College of Computer Science, Sichuan University, Chengdu 610065, China.*
- *Zhiwen Fang is with School of Biomedical Engineering, Southern Medical University, Guangzhou 510515, China.*

Corresponding Author: Le Zhang (Email:zhangleuestc@gmail.com).



(a) w/o. LA-Batch (b) w. LA-Batch

Fig. 2. Comparison between without and with the LA-Batch on the ShanghaiTech Part B dataset: x-axis represents the ground truth and y-axis represents the estimated number.

B [9] and UCF-QNRF dataset [17]. More specifically, Figure 1 demonstrates the statistics of generated image patches from these two datasets, where the data batch consists of multiple image patches during training phase. One could observe that the range of counts in patches is large and the low-density regions are overwhelmingly dominant. In consequence, such highly imbalance training data results in over/under-estimation in counting number. As shown in Figure 2(a), one could observe that MCNN [9] overestimates on sparse images while underestimating on dense images.

Based on the above observation, we propose solving the under-/over-estimation problem by introducing locality-aware concept. To be specific, we propose a simple but effective batch construction method termed **Locality-Aware Batch Construction (LA-Batch)** which serves as an alternative to the commonly used random batch construction with better generalization abilities. In brief, to jointly consider the intrinsic data structure and their ground truth, we apply local-sensitive hashing to partition the training data into different bins from which the training batches are randomly sampled . Furthermore, we propose a locality-aware data augmentation approach which employs upsampling or down-sampling to augment the training data based on the loss. Different from most existing works with delicate design of architecture, the proposed method is readily pluggable into any CNNs architecture and amenable to the end-to-end training. As shown in Figure 2(b), MCNN equipped with the proposed LA-Batch is able to deliver a better optimization approach to alleviate the overfitting problem. The contribution of this paper could be summarized as follows:

- A data-dependent batch construction method (LA-Batch) is proposed, which not only samples the candidate examples to favor diverse instances with high representation ability, but also adaptively augments the training data based on the training loss.
- The performance enhancement of LA-Batch enjoys computational efficiency. Like random batch SGD, our method with gradient computation (backpropagation) is also scalable because it only uses a small subset of all candidates. Moreover, the proposed LA-Batch is easy to implement without any sophisticated architecture design.
- LA-Batch is independent from any backbone network architectures and could serve as a complementary piece to recent improvements in crowd counting and adversarial defense. Extensive experiments on different state-of-the-art architectures and a customized one validates this advantage.

2 Related Work

Counting by regression has been extensively adopted in crowd counting thanks to their remarkable performances. To further enjoy better generalization ability in scenarios with different crowd densities, a large number of deep learning methods have been proposed, which are briefly reviewed in this section.

Context-aware CNNs Wang *et al.* [10] proposed to directly regress the total people number by adopting AlexNet [18] to extract features. To further improve the generalization ability, Fu *et al.* [19] proposed classifying the image into different density levels. In addition, a series of methods were proposed to consider a variety of local contexts such as density map [8], locality-aware features [20], [21], and Contextual Pyramid CNNs [22]. More recently, some methods are proposed to focus on producing a high-quality density map. For example, dilated kernels are used to deliver larger reception fields and to replace pooling operations in [23], [24]. Cao *et al.* [25] proposed a so-called pattern consistency loss to exploit the local correlation of density maps. There are also some methods are proposed to refine the density maps [26], [27], [28].

Scale-aware CNNs Different from Context-aware CNNs, some works proposed employing different multi-scale CNN architectures to achieve robust performance across different scenes with various perspectives, crowd densities, and scale variations. For example, Boominathan *et al.* [13] proposed a convolutional neural network consisting of both deep and shallow model for crowd counting. Similarly, Zhang *et al.* [9] proposed a multi-column CNN (MCNN) architecture, where the receptive fields of three different sizes are adopted in each individual CNN. However, these models are with a lot of training parameters which require exhaustive model pretraining and finetuning. Similar to MCNN, "Hydra CNN" [14] was proposed to estimate object densities in different crowded scenarios in a scale-aware manner. To select the optimal CNN regressor for a particular input patch, switching and growing mechanism are introduced in [15] and [29] to equip different regressors with different receptive fields, respectively. However, all these aforementioned methods inevitably increase training complexity caused by the sophisticated architectures. The deeply learned features with different receptive fields were encapsulated into a compact single vector representation amenable to efficient and accurate counting by the way of "Vector of Locally Aggregated Descriptors" (VLAD) [30]. Shen *et al.* [31] applied cross-scale consistency constraints to adversarially train a deep regression model.

Multi-task CNNs Based on the observation that a single predictor is insufficient to achieving a robust crowd counting, Kumagai *et al.* [32] proposed Mixture of CNNs (MoCNN) which employs a set of expert CNNs and a gating CNN to adaptively select the appropriate expert CNN for each input image. Walach *et al.* [33] deployed CNNs in a boosting process where the training is done in stages. DecideNet [11] proposed combining detection and regression together to learn a density map, which has shown a better generalization. Decorrelated ConvNet (D-ConvNet) [16], [34] extended the negative correlation learning into a deep neural network so that a pool of decorrelated deep regressors are learned simultaneously. Idrees *et al.* [35] proposed a composition loss that simultaneously solves the problems of counting, density map estimation and localization of people in a given dense crowd image.

Learning with Additional Data Wang *et al.* [10] augmented training data with additional negative samples whose ground truth count is set to zero. Liu *et al.* [36] proposed leveraging available unlabeled crowd imagery for crowd counting in a learning-to-rank framework. Wang *et al.*[37] used synthetic data to augment the crowd counting model via domain adaptation. In contrast with these methods, our data augmentation method is able to create new loss-adaptive training patches from the original data itself instead of borrowing other data, thus enjoying a narrower data distribution gap.

Different from most existing crowd counting methods which focus on the design of network architectures, this work contributes to the learning paradigm and the proposed method is independent from the network architecture. In other words, our method enjoys compatibility and generalization to existing deep crowd counting approaches.

3 Locality-Aware Batch Construction for Crowd Counting

As discussed in Introduction, extremely dense crowd image labeling is quite labor-intensive since each image often contains hundreds even thousands of persons, which makes difficulty in precise annotation. Such a phenomenon will make the dataset overwhelmed by less crowded images and lead to poor generalization in testing phase.

To construct locality-aware batch for training deep neural networks, we propose first partition the training data by hashing followed by loss-adaptive sampling. These two steps are elaborated in following sections.

3.1 Locality-Aware Data Partition

It is well known that deep learning is usually optimized by SGD where randomly selected samples are constructed as batches in each iteration. Compared to full gradient optimization, the stochastic batch optimization has shown advantages in handling large-scale problem since the cost of computation at each step is only proportional to the batch size. When encountering imbalance data, however, the stochastic batch is always dominated with data coming from a small subset in the target space, which may lead to a biased prediction. To simultaneously enjoy the advantages of stochastic optimization and full gradient optimization, we propose using a subset of data points as the "surrogates" of the whole dataset at each iteration. Unlike the stochastic optimization methods using random data points at each step, we pick up surrogate data points by taking the internal structure of data into consideration to construct the batch. The process of Locality-Aware Data Partition (LADP) is summarized in Algorithm 1.

We first partition the training data based on the output value (see line 1 in Algorithm 1). Specifically, let G be the number of groups, then each group is with the width of $\lceil y_{max}/G \rceil$, where y_{max} denotes the maximum value of the people counts. In consequence, the estimation people numbers in groups are sorted in ascending order and constrained into $[0, y_{max}]$. In addition, we utilize the internal structure of data for further data partition (see lines 2–3 in Algorithm 1). For each bin, we propose to use a low cost hashing method termed Locality-Sensitive Hashing (LSH) [38] to map training image patches into smaller bins such that similar image patches collapse to the same bin. In details, the hash code $h(i)$ for each training image patch $\mathbf{x}_i \in \mathcal{R}^{L\times W\times 3}$

$$h(i) \leftarrow \frac{1}{3}\sum_{c=1}^{3} {\mathbf{w}_c}^\top vec(\mathbf{x}_i(:,:,c)), \tag{1}$$

is calculated by for each data, where L, W denotes the image length and width, vec denotes the vectorization operator. The projection vector $\mathbf{w}_c \in \mathcal{R}^{LW}$ is randomly generated by Gaussian distribution for each channel c [1]. We evenly divide the group into a set of B sub-bins according to their hash values. One surrogate patch is selected from each bin as the approximation of data points collide in the same bin.

The overall loss $\tilde{\ell}$ is obtained by summing the losses of the surrogate data points $\sum_i^C \ell(\mathbf{x}_i, y_i)$, where C denotes the batch size. Then the weights of the deep model for crowd counting are updated based on the loss function $\tilde{\ell}$ with the standard back-propagation algorithm.

Algorithm 1 LSH for Data Partition

INPUT: $\{(\mathbf{x}_i, y_i)|_{i=1}^n\}$ with $\mathbf{x} \in \mathcal{R}^{L\times W\times 3}$, B, G.

1: Group the data into G group with width $\lceil y_{max}/G \rceil$
2: Generate $\mathbf{w}_c \in \mathcal{R}^{LW}$ for each channel, whose element is drawn from Gaussian Distribution.
3: Set hash value $h(i) \leftarrow \frac{1}{3}\sum_{c=1}^{3} {\mathbf{w}_c}^\top vec(\mathbf{x}_i(:,:,c))$
4: We evenly divide the group into a set of B bins according to their hash values.
5: **for** $t = 0, 1, 2, \cdots$, **do**
6: Sample one data point from each bin, compute the corresponding $\ell(\mathbf{x}_i, y_i)$ of the sampled data.
7: Compute $\tilde{\ell} \leftarrow \sum_i^C \ell(\mathbf{x}_i, y_i)$
8: Update gradient and model parameters.
9: **end for**

OUTPUT: Updated network parameters.

3.1.1 Theoretical Insight of LSH based Sampling

In this subsection, we give the theoretical analysis on LSH on error bound inspired by p-stable distribution.

We first give the definition of p-stable distribution (also termed Lévy alpha-stable distribution) as follows:

Definition 1. [39] A distribution $\mathcal{D}$ over $\mathcal{R}$ is called p-stable, if there exists $p \geq 0$ such that for any n real numbers $a_1, a_2, \cdots, a_n$ and i.i.d. variables $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n$ with distribution $\mathcal{D}$, the random variable $\sum_i a_i \mathbf{x}_i$ has the same distribution as the variable $(\sum_i |a_i|^p)^{(1/p)}\mathbf{x}$, where $\mathbf{x}$ is a random variable with distribution $\mathcal{D}$.

Specifically, a Gaussian distribution $\mathcal{D}_G$, defined by the density function $g(\mathbf{x}) = \frac{1}{\sqrt{2\pi}} e^{-x^2} dx$ is 2-stable [40].

For any p-stable distribution, the probability of any two data points collide in the same bin enjoys the following property:

Lemma 1 (Collision Probability). For any two data points $\{\mathbf{x}_i, y_i\}, \{\mathbf{x}_j, y_j\}$, let $c = \|\mathbf{x}_i - \mathbf{x}_j\|_p$, $p(u)$ denote the the probability (as a function of u) that $\mathbf{x}_i, \mathbf{x}_j$ collide for a hash function $h(\cdot)$, $f_p(t)$ be the probability distribution function of the **absolute value** of the p-stable distribution, and bin width $r \in \mathcal{R}$, then the probability of two data points under

1. For gray images, we have $h(i) \leftarrow {\mathbf{w}_c}^\top vec(\mathbf{x}_i(:,:))$.

the mapping of the p-stable distribution collide in the same bin is

$$p(u) = Pr[h(i) = h(j)] = \int_0^r \frac{1}{c} f_p(\frac{t}{c})(1 - \frac{t}{r})dt, \quad (2)$$

A direct result for Gaussian distribution following the previous lemma is expressed as:

Lemma 2 (Locality Aware). The probability of any two data points with euclidean distance c in original space under the mapping of the Gaussian distribution to collide in the same bin (bin width r) is bounded by the Gaussian integral $\frac{2}{\sqrt{\pi}} \int_0^{\frac{r}{\sqrt{2}c}} e^{-x^2} dx$

Proof. The probability distribution function of the absolute value of the Gaussian distribution (folded normal distribution [41]), i.e., $f_p(\mathbf{x})$, is defined as $f_2(\mathbf{x}) = \frac{2}{\sqrt{2\pi}} e^{\frac{-x^2}{2}}$ [42].

By plugging $f_2(\mathbf{x})$ into 2, we have

$$\begin{aligned} Pr[h(i) = h(j)] &= \int_0^r \frac{1}{c} \frac{2}{\sqrt{2\pi}} e^{-(\frac{t}{c})^2/2}(1 - \frac{t}{r})dt \quad (3) \\ &= \frac{2}{\sqrt{\pi}} \int_0^r e^{-(\frac{t}{c})^2/2} d\frac{t}{\sqrt{2}c} - \frac{1}{\sqrt{2}cr} \int_0^r e^{-(\frac{t}{c})^2/2} t dt \\ &\leq \frac{2}{\sqrt{\pi}} \int_0^r e^{-(\frac{t}{c})^2/2} d\frac{t}{\sqrt{2}c} \\ &= \frac{2}{\sqrt{\pi}} \int_0^{\frac{2}{\sqrt{2}c}} e^{-x^2} dx \end{aligned}$$

□

Remark 1. LSH is locality-aware in context of counting, since 1) the patches are sampled from different locations in the original image that are used in constructing bins; 2) the geometric property remains after hashing. Specifically, when the two original datapoints distance $c \to 0$, the probability of collision is almost 1, which means when two data points are close to each other, they will collide in the same bin with probability of 1. The Gaussian integral bound gives us the intuition the nearer two points in original space, they will be hashed into the same with higher probability, which explains why our proposed LADP is locality-aware. Furthermore, r is one of the key factor to decide to whether two data points collide in the same bin. If $c \gg r$, the two data points are unlikely to be hashed into same bin. Different from other sampling strategies such as random sampling or balanced sampling that only output space y is considered, our proposed LADP also incorporates intrinsic structure of input space for batch construction. This essentially lessens the difficulty of the optimization, which is verified by our following experiments.

3.2 Locality-Aware Data Augmentation

As shown in Figure 1, the imbalance and complex data distribution has been one of major factors to hinder the performance improvement of crowd counting. To solve this problem, we further propose a Loss-Aware Data Augmentation (LADA) approach to conduct data augmentation so that the diversity of training data could be increased and the data imbalance issue is alleviated. Furthermore, most state-of-the-art deep architectures are trained with stochastic batch mode, which means that the training batch is dominated by data coming from a small subset in target space as illustrated in Figure 1. As a result, the model prediction favors the majority of training outputs. To alleviate such a training bias, we propose zoom-in/-out strategy on each individual training patch for data augmentation. For the over-estimation cases, we aim at increasing training data with small outputs, i.e., less crowded image patches. For the under-estimate cases, we aim at generating training data with large outputs, i.e., more crowded image patches. Specifically, we first calculate the residual e^t of data $\mathbf{x}$ for t-th epoch,

$$e^t = \tilde{y}^t - y, \quad (4)$$

where $\tilde{y}^t$ is the prediction at the t-th epoch and y refers to the ground truth of people counting. The scale factor α^t is defined as follows,

$$\alpha^t = e^t / y. \quad (5)$$

Note that, α^t is normalized according to the ground truth value. We further define the zoom rate function $g(\alpha^t)$ as follows,

$$r^t = g(\alpha^t) = 1 + \lambda(\frac{1}{1 + \exp \gamma\alpha^t} - 0.5) \quad (6)$$

where the parameters $0 < \lambda < 2$ and $\gamma > 0$ are used to control the scale. The function is visually illustrated in Figure 3. From the result, one could observe when $\alpha^t > 0$ (i.e., the residual $\ell^t > 0$), the zoom rate $r^t < 1$. In other words, if the prediction is overestimated, zoom-in strategy is adopted to generate less crowded image patches. When $\alpha^t < 0$ (i.e., the residual $\ell^t < 0$), the zoom rate $r^t > 1$. In other words, if the prediction is overestimated, the zoom-out strategy is used to generate more crowded image patches.

With the above settings and definitions, a new zoomed image $\mathbf{x}^{t+1}$ at the t-th epoch will be generated as below:

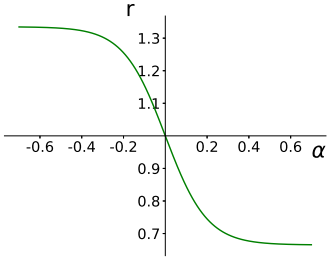


Fig. 3. Zooming function $r^t = g(\alpha^t)$, where $\lambda = 2/3$ and $\gamma = 10$.

$$\mathbf{x}^{t+1} = f(\mathbf{x}^t, r^t), \quad (7)$$

where f denotes the new image generation process with the zoom rate r^t. The procedure is illustrated in Figure 4. With the above generation, one could obtain a lot of new data to alleviate the aforementioned data imbalance issue.

4 Experiments

We evaluate our proposed method on five different datasets of which four are publicly accessed. Different from most existing DNN based methods, the proposed approach is simple and could be regarded as an addon which is complementary and able to improve most existing architectures as verified in the following experiments.

The Network Architecture To demonstrate the effectiveness of

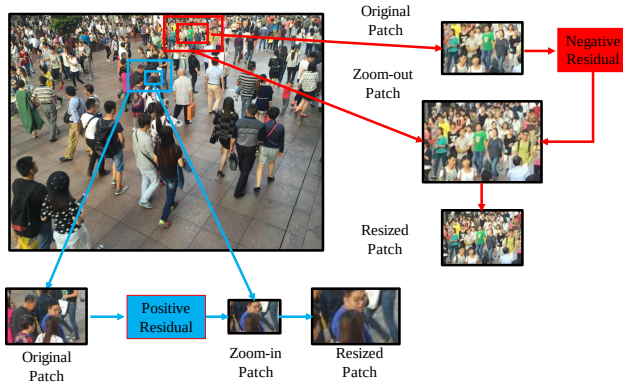


Fig. 4. Locality-Aware Data Augmentation

the proposed LA-Batch, we simply employ a pre-trained VGG-16 network with small modifications. More specifically, the stride of the fourth max-pool layer is set to 1, and the fifth pooling layer was removed. As a result, a much larger feature map with richer information is given. To handle the receptive-field mismatch caused by the removal of stride in the fourth max-pool layer, we duplicate the receptive field of convolutional layers after the fourth max-pool layer by using the technique of holes introduced in [43]. Furthermore, we apply group convolution on the output feature map to obtain the density map. For those datasets without providing density map, we just follow the protocol in [9] to compute the ground-truth density map for regression. In order to show the versatility of the proposed method, we also apply LA-Batch to the recently proposed CRSNet [23].

Our method is implemented in Tensorflow on a workstation with a TitanX GPU. The Adam optimizer is used in the proposed network with a mini-batch size of 60. We use an exponential learning rate decay with a initial learning rate of 0.0001. The learning rate decays every 5 epochs with a decay rate of 0.95. For all the dataset, we also report the results from our baseline method in which both LADP and LADA are disabled. We name it as "Base Model".

There are two parameters in the LADP, i.e., the number of group G and the number of bin B. We empirical find that satisfactory performance is obtained with $G = 6, B = 10$. In LADA, there are also two parameters in the zooming function to control scale of zooming, i.e., λ and γ. We empirical find that $\lambda = 2/3, \gamma = 10$ could give satisfactory performance.

The Evaluation Metric The widely used mean absolute error (MAE) and the root mean squared error (RMSE) are adopted to evaluate the performance of all the tested approaches. The MAE and RMSE are defined as follows:

$$\begin{aligned} MAE &= \frac{1}{N}\sum_{i=1}^{N}|(y_i - \tilde{y}_i)|, \\ RMSE &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}|y_i - \tilde{y}_i|^2}, \end{aligned} \tag{8}$$

where N denotes the total number of testing images, y_i and $\tilde{y}_i$ are the ground truth and the estimated value for the i-th image, respectively. The averaged results of our methods are reported over five runs with different random projection vectors in LADP.

The Mall Dataset The Mall dataset [44] contains 2000 images collected from a shopping mall. Each image is with a fixed resolution of 640×480. We follow the predefined settings to use the first 800 frames as the training set and the rest 1200 frames for testing. The validation set contains 100 images randomly selected from the training set. The evaluation results are exhibited in Table 1.

From the results, we observe that LA-Batch is able to beat most of state-of-the-art methods even though only a very simple architecture is used. Note that, our method performs the best even though it neither uses the ensemble scheme employed by the "MoCNN" and "Boosting CNN" methods, nor couples with detection networks like [11].

TABLE 1
Crowd Counting Performance Comparison on the Mall dataset.

Method	MAE	RMSE
Count Forest [45]	4.40	2.40
Exemplary Density [46]	1.82	2.74
Boosting CNN [33]	2.01	-
MoCNN [32]	2.75	13.40
Weighted VLAD [47]	2.41	9.12
DecideNet [11]	1.52	1.90
Base Model	1.87	2.85
Base +LA-Batch	1.60± 0.4	1.95± 0.5
CSRNet [23]	1.70	2.03
CSRNet +LA-Batch	**1.34**± 0.2	**1.60**± 0.3

The Shanghaitech Dataset The Shanghaitech dataset [9] is a large-scale crowd counting dataset which consists of two parts. To be exact, Part A includes 482 images which are randomly captured from the Internet, and Part B includes 716 images that are taken from the busy streets in Shanghai. Each part is divided into training and testing subset. The crowd density significantly varies among the subsets, making difficulty in estimating the number of pedestrians. We compare our method with five latest deep regression methods on this dataset. All the detailed results for the methods are illustrated in Table 2. In the same way, one could see that our proposed method is superior to all the baselines. Note that compared to Part B, our performance improvement on part A is much smaller since the simple base model may be insufficient to model the complex and diverse images in part A. Nevertheless, our proposed method with a simple base architecture is still able to deliver better results compared to those state of the arts.

TABLE 2
Crowd Counting Performance Comparison on the ShanghaiTech dataset.

	Part A		Part B	
Method	MAE	RMSE	MAE	RMSE
Zhang *et al.*. [8]	181.8	277.7	32.0	49.8
MCNN [9]	110.2	173.2	26.4	41.3
Switch-CNN [48]	90.4	135.0	21.6	33.4
D-ConvNet [16]	73.5	112.3	18.7	26.0
DecideNet [11]	-	-	21.53	31.98
ACSCP [31]	75.7	**102.7**	17.2	27.4
SANet [25]	67.0	104.5	**8.4**	**13.6**
IG-CNN [29]	72.5	118.2	13.6	21.1
Base Model	105.4	140.5	22.7	37.8
Base + LA-Batch	74.2± 1.2	114.0± 4.0	14.8± 0.4	22.5± 1.4
CSRNet [23]	68.2	115.0	10.6	16.0
CSRNet + LA-Batch	**65.8**± 0.9	103.6± 2.7	8.6± 0.4	14.0± 0.6

The WorldExpo dataset The WorldExpo'10 dataset [8] includes 1132 annotated video sequences collected from the World Expo 2010 event. The training set consists of 3380 frames and the

rests are used for testing. Since the Region Of Interest (ROI) are provided for five test scenes (S1–S5), we follow the evaluation protocol used in [8] which only counts the persons within the ROI area. We also employ the MAE as suggested by [8] for evaluation. The results of our proposed approach on all tested scenes and the comparisons to other methods are summarized in Table 3. From the results, one could find that the proposed method is competitive with the evaluated well-established baselines even though our method does not utilize the perspective information. Furthermore, another advantage of our method over them is that our method performs more stable.

The UCF_CC_50 Dataset The UCF_CC_50 dataset [49] contains 50 images that are randomly collected from the Internet. Different from the above datasets, this dataset consists of many extremely dense crowd images with the average of 1280 heads per image. The dataset is challenging due to large variations in head number among different images from a small number of training images. The results of state-of-the-art performances are summarized in Table 4. One could observe that introducing switch mechanism and ensemble learning into CNNs is able to significantly reduce estimation error for such dense situations. Note that this dataset only consists of 50 images, which is much smaller than other used benchmark datasets. Nevertheless, with our data augmentation, our method is able to advance a new state-of-the-art performance with a simple architecture.

The UCF-QNRF Dataset The UCF-QNRF dataset [17] consists of 1,535 jpeg images with 1,251,642 people in them. The training set is made of 1,201 of these images. Unlike in ShanghaiTech, there are dramatic variations both in crowd density and image resolution. We summarize the results of various methods in Table 5. In this dataset, our proposed method shows inferior results compared to Bayesian+ [28] and SFCN+ [37]. Nevertheless, we want to point out that the backbones used in their methods are much better than ours. For example, CSRNet only achieves MAE of 148, while backbone models in Bayesian+ and SFCN+ can achieve MAE of 106.8 and 114.8, respectively. In addition, SFCN+ utilizes additional simulation dataset to boost up performance.

5 Ablation Study and Analysis

In this section, we conduct ablation study to further under stand the merits of LA-Batch from various aspects. We choose ShanghaiTech B dataset as our default studying dataset for analysis.

Comparing with Other Optimization Alternatives Indeed we are not the first to study the data imbalance problem and existing solutions can be found in several other tasks such as object recognition, object detection, semantic segmentation and so on. Most existing learning algorithms produce inductive bias (learning bias) towards the frequent (majority) classes if training data are not balanced, resulting in poor minority class recognition performance. A simple approach to alleviating data imbalance in model learning is to re-sample the training data (a pre-process), e.g. by offline data augmentation [52], balanced sampling [52]. To further demonstrate the effectiveness of LA-Batch, we further compared our proposed method with more batch construction strategies proposed for other tasks: 1) Random sampling (all the data is randomly sampled to construct the training batches.) 2) Offline data augmentation (we offline build an uniform distributed training set from scaled image patches and then apply random sampling to train a model.) 3) Balanced Sampling (We group the training data based on the output value y). 4) Adaptive samplingwe adaptively sample the data samples in each bash according to their loss without zoom-in and zoom-out strategy. To our best knowledge, the fourth baseline has not been systematically studied in existing literature. We also include the proposed LADA and LADP strategy for comparison. The averaged results over five runs on the ShanghaiTech B are summarized in Table 6.

From the results, we observe that the following phenomenon: (1) All the strategies could improve the performance over random sampling. (2) Offline data augmentation, which is similar to LADA in an offline manner, performs worse than LADA. (3) Balanced sampling, which resembles LADP without hashing in an online manner, is outperformed by LADP. (4) Adaptive sampling, which is similar to LADA without online adaptive augmentation, leads to inferior results than LADA. (4) LA-Batch, which integrates LADA and LADP, performs the best. (5) Our proposed methods is generic and could be readily pluggable into different CNN backbones.

The Effect of Locality-Aware Data Partition In this experiment, we further show how the Locality-Aware Data Partition (LADP) affects the deep regression optimization in terms of MAE. The performance is evaluated by averaging MAE of training patches in each iteration. We report the comparison of training with and without LADP on the training data in Figure 5(a). From the result, we observe that by deploying LADP, the model training process is more stable than the fully random batch construction. In addition, training with LADP is able to achieve lower MAE for training patches.

In LADP, Locality-sensitive hashing (LSH) is used which considers the original geometric property when conduct the data partition. Lemma 2 also proves that LSH is able to distribute the similar data in the same bin and dissimilar data in different bins with high possibility. To further reveal this locality-aware benefits in optimization, we show experimental results in Figure 5(b) by comparing LSH with random hashing regarding the number of samples (indicated by the bars) and mean pairwise log distance (represented by the curves) for each bin. We observe that 1) In contrast with LSH, random hashing almost evenly distributes the data; 2) The average pairwise distance in each group is much smaller for LSH than that of the random hashing which means LSH is able to group similar patches together. All these experimental findings could support the claims in Lemma 2.

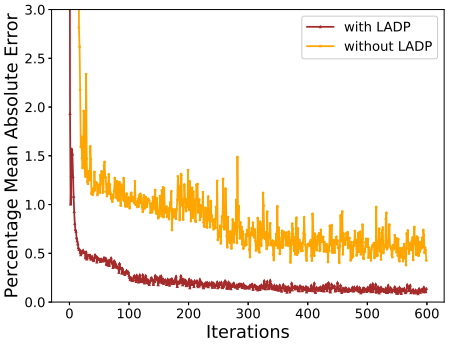
To further understanding how LADP works, we give examples of training batch from ShanghaiTech B in Figure 6. The first row and second row show the patches in a training batch with LADP and without LADP respectively. We observe that the batch constructed with LADP are in better diversity than that without LADP. Specifically, the second row demonstrates that most of patches are from floor background regions without help of LADP. In contrast, LADP is able to create training batches consisting of more diverse patches of different local regions and people densities.

The Effect of Locality-Aware Data Augmentation In this experiment, we further analyze the Locality-Aware Data Augmentation (LADA) from the perspective of training patch distribution. From Figure 7, one could observe that LADA makes the distribution more focus on regions with larger errors (e.g., patches with less than 20 and more than 70 people [2]) by generating more image patches on the training data. Note that LADA is a loss-driven

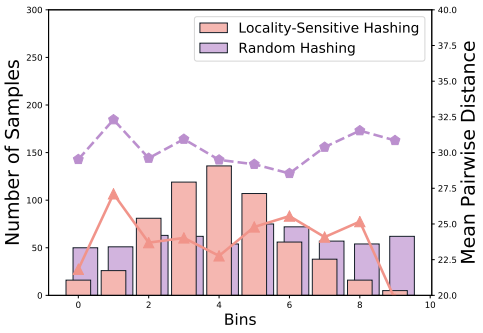
2. See Fig. 2 in introduction.

TABLE 3
Crowd Counting Performance Comparison on the Expo dataset.

Method	S1	S2	S3	S4	S5	Ave MAE
Zhang *et al.*. [8]	2.00	29.50	9.70	9.30	3.10	12.90
MCNN [9]	3.40	20.60	12.90	13.00	8.10	11.60
Switch-CNN [48]	4.40	15.70	10.00	11.00	5.90	9.40
D-ConvNet [16]	**1.9**	12.1	20.7	8.3	2.6	9.1
DecideNet [11]	2.00	13.14	8.90	17.40	4.75	9.23
ACSCP [31]	2.8	14.05	9.6	**8.1**	2.9	**7.5**
SANet [25]	2.6	13.2	9.0	13.3	3.0	8.2
ic-CNN [27]	17.0	12.3	9.2	8.1	4.7	10.3
DRSAN [26]	2.6	11.8	10.3	10.4	3.7	7.76
Base Model	3.8	15.6	18.2	14.3	6.9	11.8
Base+ LA-Batch	3.1± 0.2	12.0± 0.5	16.1± 0.7	10.8± 0.4	**2.4**± 0.1	8.9± 0.4
CSRNet [23]	2.9	11.5	8.6	16.6	3.4	8.6
CSRNet+LA-Batch	2.4± 0.1	**11.0**± 0.3	**8.1**± 0.5	13.5± 0.2	2.7± 0.2	**7.5**± 0.3



(a) Training Curve



(b) LSH vs Random Hashing

Fig. 5. Analysis of Locality-Aware Data Partition. Figure 5(a) shows training curve comparison between with and without LADP. The bars in Figure 5(b) shows the statistics of the bin partition and the curve represents the mean pairwise log distance in each bin of LSH and random hashing.

Fig. 6. Examples of patches in a training batch: The first row and second row show the patches in a training batch with LADP and without LADP respectively. With LADP, the image patches are more diverse in both the number of people and background.

TABLE 4
Crowd Counting Performance Comparison on the UCF_CC_50 dataset.

Method	MAE	RMSE
Density learning [1]	493.4	487.1
FHSc+MRF [49]	419.5	487.1
Zhang *et al.*. [8]	467.0	498.5
CrowdNet [13]	452.5	-
Hydra2s [14]	333.73	425.3
MCNN [9]	377.6	509.1
D-ConvNet [16]	288.4	404.7
Switch-CNN [48]	318.1	439.2
ACSCP [31]	291.0	404.6
IG-CNN [29]	291.4	349.4
DRSAN [26]	219.2	250.2
ic-CNN [27]	260.9	365.5
SANet [25]	258.4	334.9
Bayesian+ [28]	229.3	308.2
SFCN+ [37]	214.2	318.2
Base Model	330.2	460.3
Base +LA-Batch [23]	270.5± 3.2	410.2± 8.5
CSRNet [23]	221.6	304.9
CSRNet+LA-Batch	**203.0**± 2.0	**230.6**± 6.4

TABLE 5
Crowd Counting Performance Comparison on the UCF-QNRF dataset.

Method	MAE	RMSE
MCNN [9]	227	462
Switch-CNN [48]	228	445
CMTL [50]	252	514
CL [51]	132	191
Bayesian+ [28]	**88.7**	**154.8**
SFCN+ [37]	102.0	171.4
Base Model	163	328
Base +LA-Batch	137± 1.8	230± 6.2
CSRNet [23]	148	313
CSRNet +LA-Batch	113± 1.9	210± 7.5

data augmentation approach and it does not force the training data distribution to be an exactly uniform distribution. In addition, it could smartly alleviate the distribution bias in the training process.

Parameter Analysis The proposed method has four tunable parameters : the number of groups G and bins B in LADP, and two parameters in the zooming function to control scale of zooming, i.e., λ and γ in LADA. In this section, we conduct a series of experiments to study the sensitivity issues of the parameters. The default setting for parameters is $G = 6, B = 10, \lambda = 2/3, \gamma = 10$ and we study the sensitivity of LA-Batch with $G \in [1, 4, 6, 12], B \in [1, 5, 10, 20], \lambda \in [1/10, 1/3, 2/3, 4/3], \gamma \in$

TABLE 6
Compare LA-Batch with Other Optimization Alternatives on Different Backbones.

	Base Model		CSRNet	
Method	MAE	RMSE	MAE	RMSE
Random Sampling	22.7	37.8	10.6	16.0
Offline Data Augmentation	18.4	26.5	10.1	15.2
Balanced Sampling	19.5	29.2	10.3	15.8
Adaptive Sampling	17.5	25.1	9.7	14.9
LADA	16.8	25.3	9.4	14.5
LADP	18.0± 0.9	27.2± 1.8	10.2± 0.7	15.3± 0.8
LA-Batch	**14.8**± 0.4	**22.5** ± 1.4	**8.6**± 0.4	**14.0**± 0.6

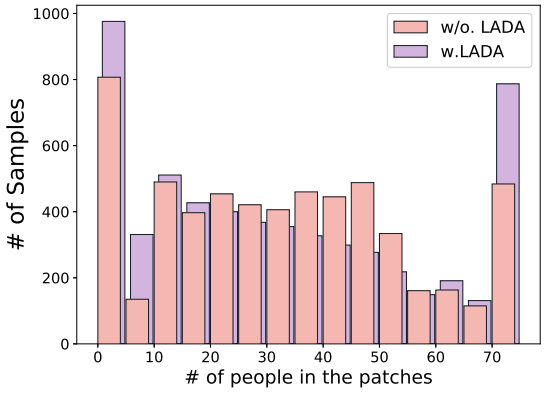


Fig. 7. Training Distribution Comparison between w/o. and w. LADA.

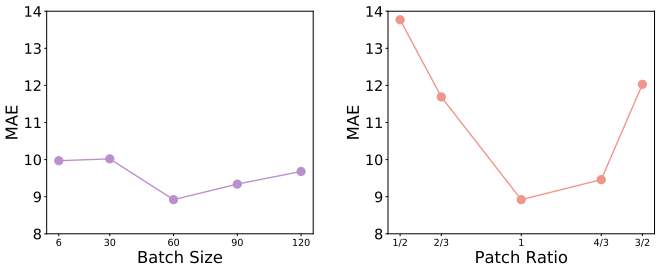
$[1, 5, 10, 20]$. Results are summarized in Table 7 . We empirical find that the performance is less sensitive for B, λ, γ than G.

TABLE 7
Parameter Analysis on ShanghaiTech B data.

G	MAE/RMSE	B	MAE/RMSE
1	14.26/17.57	1	9.97/18.67
4	10.35/16.25	5	10.02/21.23
6	8.92/13.71	10	8.92/13.71
12	9.19/14.91	20	9.68/17.92
λ	MAE/RMSE	γ	MAE/RMSE
1/10	9.78/17.77	1	9.59/15.52
1/3	9.96/18.01	5	9.50/15.05
2/3	8.92/13.71	10	8.92/13.71
4/3	9.70/16.55	20	9.85/19.97

Effect of Batch Size and Patch Size. In this experiment, we analyze the effect of batch size and patch size on the ShanghaiTech Part B dataset. The default setting of batch size is 60 and the batch size varies in the range of $\{6, 30, 60, 90, 120\}$. The experiment results is shown in Figure 5. The default patch size is 180×296. In the experiment of the patch size analysis, we treat the default width as the baseline marked by 1.0 and only vary width in the range of $\{1/2, 2/3, 1, 4/3, 3/2\}$ by keeping the original aspect ratio of images. The experiment results are shown in Figure 5. We empirically found that the proposed methods are not very sensitive to those hyper-parameter settings. Interestingly, we could also see that the proposed methods show better robustness to the batch size when compared with the patch size.

Time Efficiency. The proposed methods usually yield improved performance with negligible computational over-heads. To demonstrate this, we report the training time per epoch for different



(a) Effect of Batch Size. (b) Effect of Patch Size.

Fig. 8. Effect of Batch and Patch size.

methods in Table 8. From the results, we observe that the proposed method only brings 6% extra training time.

TABLE 8
Training Time of Different Methods per Epoch.

Methods	CSRNet [23]	+LADP	+LADA	+LA-Batch
Time	237.92s	242.42s	247.66s	252.85s

6 Application to Adversarial Defense

Data imbalance problem arises in many machine learning tasks where the proposed LA-Batch could be valuable. To show its versatility, we apply LA-Batch to adversarial defense. Neural networks have been shown to be vulnerable to adversarial perturbations. Specifically, adversarial perturbations are imperceptible and can cause a significant drop in the classification accuracy. The level of distortion is measured by the margin between the original and the perturbed input. Formally, the margin of a data point is defined as the minimum distance that a data point has to be perturbed to change the classifier's prediction. Thus, the larger the margin is, the more robust the classifier is w.r.t. this input. Adversarial training, which essentially minimizes the maximum loss within a fixed perturbation ϵ on the training data using projected gradient descent (PGD)/Fast Gradient Sign Method (FGSM) [53], is one of the most popular methods for adversarial defense. Unfortunately, a fixed perturbations only cater the majority of training samples by sacrificing the contribution of minorities. From Figure 9, we observe that margin distribution of samples is highly imbalanced, which indicates that adversarial training [53] with the commonly-used setting of fixed margins yields small margins for the the majority of data points after convergence. This is not optimal as in practice those data with small margins could be easily attacked by existing adversarial attack methods.

To address the same data imbalance challenge, we adopt the same locality-aware concept and propose a locality-aware adversarial training (adversarial training + LA-Batch) based on the adversarial training[53]. The algorithm is summarized in the appendix. It first calculates the margin ϵ_i for data point $(\mathbf{x}_i, y_i)$ with given classification model $F_\Theta(\cdot)$ that is parametrized by Θ. Then we apply LADP to group and hash training data $\{(\mathbf{x}_i, y_i)\}$ according to ϵ_i[3]. After constructing the training batch with LADP, we follow [53] and use PGD to generate the the corresponding adversarial examples $\{\mathbf{x}_i^{adv}\}$. Then one step gradient update on

3. Note that here LADP is applied based on the value of ϵ_i rather than y_i, which is different from that usage of LADP in crowd counting.

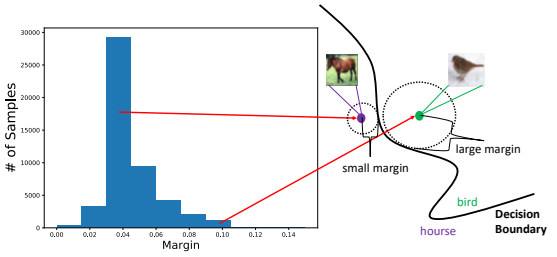


Fig. 9. Highly Imbalanced Margins in Training Data for Adversarial Training on Cifar10: The majority of samples congest on the decision boundary with small margins after adversarial training.

Θ is performed with training set $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_i^{adv}, y_i)$. With updated Θ, we conduct LADA to generate new adaptive margin ϵ_i for each datapoint.

More specifically, we first define the confidence vector of K-class classifier, i.e, $\mathbf{z} = (z_1, z_2, \cdots, z_K) = F_\Theta(\mathbf{x})$. Then the predicted label of $\mathbf{x}$ is given by $\hat{y} = \arg\max_j z_j$. The scale factor α is by $\alpha = z_y - \max_{j \neq y}(z_j) - \delta$, where z_y is the confidence score of label y and δ is the predefined confidence threshold [4]. The zoom function remains as the same in crowd counting, $r = g(\alpha) = 1 + \lambda(\frac{1}{1+\exp\gamma\alpha} - 0.5)$. Finally, the margin is adjusted with the zoom rate, i.e., $\epsilon^* = r\epsilon$. With new margin ϵ^*, we are able to generate new adversarial examples $\mathbf{x}^{adv}$. Detailed procedures are summarized in the Algorithm 2.

We follow the same experimental protocol in [53] and use the ResNet as the backbone classifier. We compare locality-aware adversarial training with different adversarial training algorithms on the STL10 [54] and Cifar10 datasets which are commonly used in existing works.

As for the evaluation metric on the robustness, we also adopt the widely used accuracy over the adversarial examples generated by certain attack methods [55]. For a certain attack method $\mathcal{A}_\epsilon(\mathbf{x})$ with the perturbation budget ϵ, the robustness is evaluated as

$$R(F_\Theta,\ \mathcal{A}_\epsilon) = \frac{1}{M}\sum_{i=1}^{M} \mathbb{1}\left[F_\Theta(\mathcal{A}_\epsilon(\mathbf{x}_i)) = y_i\right]. \qquad (9)$$

The above equation calculates the accuracy of a given model $F_\Theta(\mathbf{x})$ on the adversarial examples crafted by a given adversarial attack method $\mathcal{A}_\epsilon(\mathbf{x})$ with perturbation budget ϵ. The robustness are evaluated under FGSM attack with $\epsilon = 8/255$, and PGD attack with $\epsilon = 16/255$ [53].

We first report the results of the proposed method on STL10 dataset in Table 9. We compare our proposed method with three recent popular defense baselines, Pixel Deflection [56], Random Padding Resizing [57], Adversarial Training [53]. Among these baselines, adversarial training performs the best in terms of robustness Our results clearly show the superiority of the locality-aware strategy for this task. In order to further understand the merits of the proposed method, we investigate the individual contribution of LADP and LADA in Table 9. From the results, we observe that both LADA and LADP are effective in improving the final performance and those improvement are complementary.

After exploring the LA-Batch on the STL10 dataset and establishing the effectiveness of both LADA and LADP, we summarize the results obtained by our method and compare it with existing methods on Cifar10 dataset in Table 10. Results show that LA-Batch improves adversarial training, as expected.

To further understand the effectiveness of LA-Batch to adversarial training, we also show the margin distribution before and after training on STL10 in Figure 10. Our proposed method enlarges margins of all training points, while original adversarial training might fail to enlarge margins for points with initial margins smaller than the predefined margin ϵ. The similar observations could also be found in the testing data, which leads better performances of locality-aware adversarial training.

4. Different from α defined in crowd counting where a zero threshold is implicitly used, here we used a predefined threshold to further penalize those datapoints that commit mistakes or near the decision boundary.

Algorithm 2 Locality-Aware Adversarial Training

Require: Training set $\{(\mathbf{x}_i, y_i)\}$; a model $F_\Theta(\cdot)$ with loss function $\nabla_\mathbf{x}\ell(F_\Theta(\mathbf{x}), y)$.

Randomly initialize the parameter Θ of model $F_\Theta(\cdot)$.
$\epsilon_i = \text{CALCULATE\_MARGIN}(\mathbf{x}_i, y_i, F_\Theta(\cdot))$
Apply LADP to group and Hash $\{(\mathbf{x}_i, y_i)\}$ according to ϵ_i.
repeat
  $\mathbf{x}_i^{adv} = \text{PGD}(\mathbf{x}_i, y_i, \epsilon_i, F_\Theta(\cdot))$
  Perform one step gradient update on Θ with training set $(\mathbf{x}_i, y_i)$ and $(\mathbf{x}_i^{adv}, y_i)$.
  $\epsilon_i = \text{LADA}(\mathbf{x}_i^{adv}, y_i, \epsilon_i, F_\Theta(\cdot))$
until meet training stopping criterion

Ensure: The trained model $C(\Theta, x)$.

function CALCULATE_MARGIN($\mathbf{x}, y, F_\Theta(\cdot)$)
  $\epsilon = 0$
  $\hat{g} = \text{sign}(\nabla_\mathbf{x}\ell(F_\Theta(\mathbf{x}), y))$
  repeat
    increase ϵ
  until $C(\Theta, x + \epsilon\hat{g}) \neq y$ **return** ϵ^*
end function
function PGD($\mathbf{x}, y, \epsilon, F_\Theta(\cdot)$)
  Given Iterations T, step size β
  $\mathbf{x}_0^{adv} = \mathbf{x}$
  for $t = 1$ to $T - 1$ **do**
    $\mathbf{x}_t^{adv} = \mathbf{x}_{t-1}^{adv} + \beta\text{sign}(\nabla_\mathbf{x}\ell(F_\Theta(\mathbf{x}_{t-1}^{adv}), y))$
    $\mathbf{x}_t^{adv} = \text{clamp}(\mathbf{x}_t^{adv}, \epsilon)$
  end for
  return x_{T-1}^{adv}
end function
function LADA($\mathbf{x}^{adv}, y, \epsilon, F_\Theta(\cdot)$)
  Given λ, γ, and confidence threshold δ
  Confidence $\mathbf{z} = F_\Theta(\mathbf{x}^{adv})$
  $\alpha = z_y - \max_{j \neq y}(z_j) - \delta$
  $r = 1 + \lambda(\frac{1}{1+\exp\gamma\alpha} - 0.5)$
  $\epsilon^* = r\epsilon$ **return** ϵ^*
end function

TABLE 9
Robustness Evaluation under white-box attacks on STL10 dataset. The white-box attacks are end-to-end FGSM attack with $\epsilon = 8/255$ and PGD attack with $\epsilon = 16/255$

	Accuracy	Robustness	
Defense Method	**Clean Image**	**FGSM-8/255**	**PGD-16/255**
Pixel Deflection [56]	0.656	0.211	0.052
Random Padding Resizing [57]	0.706	0.410	0.075
Adversarial Training [53]	0.631	0.445	0.515
Adversarial Training + LADA	0.650	0.452	0.535
Adversarial Training + LADP	0.633 ± 0.004	0.455± 0.017	0.520± 0.003
Adversarial Training + LA-Batch	0.661 ± 0.003	**0.466± 0.007**	**0.549± 0.001**

TABLE 10
Robustness Evaluation under white-box attacks on Cifar10 dataset. The white-box attacks are end-to-end FGSM attack with $\epsilon = 8/255$ and PGD attack with $\epsilon = 16/255$

	Accuracy	Robustness	
Defense Method	**Clean Image**	**FGSM-8/255**	**PGD-16/255**
Pixel Deflection [56]	0.789	0.340	0.320
Random Padding Resizing [57]	0.894	0.479	0.379
Adversarial Training [53]	0.767	0.566	0.506
Adversarial Training + LA-Batch	0.780± 0.002	**0.614± 0.010**	**0.537± 0.005**

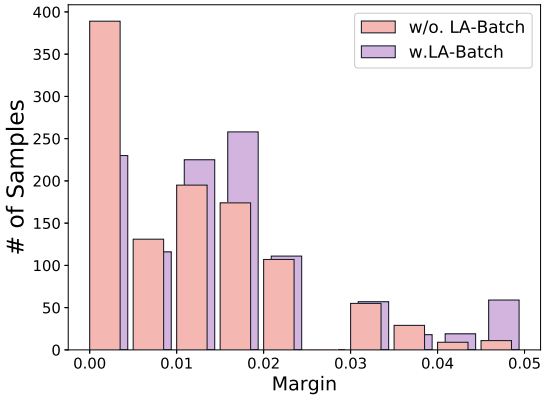


Fig. 10. Margin Distribution Comparison between w/o. and w. LA-Batch after training. Larger margins indicate more robustness.

7 Conclusion

In this paper, we investigated the problem of under-estimation and over-estimation in crowd counting. To the end, a simple but effective batch construction method, called Locality-Aware Batch (LA-Batch), was proposed to achieve generalizable features. Thanks to the independence of the proposed method with the backbone network architectures, our method could be plugged into most existing deep crowd counting methods to boost their performance in an end-to-end manner. Extensive experiments indicate the superiority of our method over several state-of-the-art baselines. The proposed LA-Batch is generic and could be valuable in other data imbalance applications and we demonstrate the versatility of LA-Batch in adversarial defense.

References

[1] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010, pp. 1324–1332. 1, 7

[2] A. B. Chan, Z.-S. J. Liang, and N. Vasconcelos, "Privacy preserving crowd monitoring: Counting people without people models or tracking," in *roceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–7. 1

[3] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and bayesian regression," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2160–2177, 2012. 1

[4] D. Ryan, S. Denman, S. Sridharan, and C. Fookes, "An evaluation of crowd counting methods, features and regression models," *Computer Vision and Image Understanding*, vol. 130, pp. 1–17, 2015. 1

[5] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 247–266, 2007. 1

[6] T. Zhao, R. Nevatia, and B. Wu, "Segmentation and tracking of multiple humans in crowded environments," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 7, pp. 1198–1211, 2008. 1

[7] 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 *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4. 1

[8] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 833–841. 1, 2, 5, 6, 7

[9] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 589–597. 1, 2, 5, 7

[10] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015, pp. 1299–1302. 1, 2, 3

[11] J. Liu, C. Gao, D. Meng, and A. G. Hauptmann, "Decidenet: Counting varying density crowds through attention guided detection and density," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 1, 2, 5, 7

[12] Q. Wang, J. Wan, and Y. Yuan, "Deep metric learning for crowdedness regression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2633–2643, 2018. 1

[13] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: a deep convolutional network for dense crowd counting," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 640–644. 1, 2, 7

[14] D. Onoro-Rubio and R. J. López-Sastre, "Towards perspective-free object counting with deep learning," in *European Conference on Computer Vision*. Springer, 2016, pp. 615–629. 1, 2, 7

[15] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, no. 3, 2017, p. 6. 1, 2

[16] Z. Shi, L. Zhang, L. Yun, X. Cao, Y. Ye, M.-M. Cheng, and G. Zheng, "Crowd counting with deep negative correlation learning," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018. 1, 2, 5, 7

[17] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 532–546. 2, 6

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances In Neural Information Processing Systems*, 2012, pp. 1097–1105. 2

[19] M. Fu, P. Xu, X. Li, Q. Liu, M. Ye, and C. Zhu, "Fast crowd density estimation with convolutional neural networks," *Eng. Appl. Artif. Intell.*, vol. 43, no. C, pp. 81–88, Aug. 2015. [Online]. Available: http://dx.doi.org.ezlibproxy1.ntu.edu.sg/10.1016/j.engappai.2015.04.006 2

[20] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, and C. Sun, "Crowd counting via weighted vlad on dense attribute feature maps," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016. 2

[21] C. Shang, H. Ai, and B. Bai, "End-to-end crowd counting via joint learning local and global count," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1215–1219. 2

[22] V. A. Sindagi and V. M. Patel, "Generating high-quality crowd density maps using contextual pyramid cnns," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2017, pp. 1879–1888. 2

[23] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1091–1100. 2, 5, 7, 8

[24] D. Deb and J. Ventura, "An aggregated multicolumn dilated convolution network for perspective-free counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 195–204. 2

[25] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750. 2, 5, 7

[26] L. Liu, H. Wang, G. Li, W. Ouyang, and L. Lin, "Crowd counting using deep recurrent spatial-aware network," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 849–855. 2, 7

[27] V. Ranjan, H. Le, and M. Hoai, "Iterative crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 270–285. 2, 7

[28] 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, 2019, pp. 6142–6151. 2, 6, 7

[29] D. Babu Sam, N. N. Sajjan, R. Venkatesh Babu, and M. Srinivasan, "Divide and grow: capturing huge diversity in crowd images with incrementally growing cnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3618–3626. 2, 5, 7

[30] Z. Shi, L. Zhang, Y. Sun, and Y. Ye, "Multiscale multitask deep netvlad for crowd counting," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 11, pp. 4953–4962, 2018. 2

[31] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd counting via adversarial cross-scale consistency pursuit," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5245–5254. 2, 5, 7

[32] S. Kumagai, K. Hotta, and T. Kurita, "Mixture of counting cnns: Adaptive integration of cnns specialized to specific appearance for crowd counting," *arXiv preprint arXiv:1703.09393*, 2017. 2, 5

[33] E. Walach and L. Wolf, "Learning to count with cnn boosting," in *European Conference on Computer Vision*. Springer, 2016, pp. 660–676. 2, 5

[34] L. Zhang, Z. Shi, M.-M. Cheng, Y. Liu, J.-W. Bian, J. T. Zhou, G. Zheng, and Z. Zeng, "Nonlinear regression via deep negative correlation learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2

[35] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 08 2018. 2

[36] X. Liu, J. van de Weijer, and A. D. Bagdanov, "Leveraging unlabeled data for crowd counting by learning to rank," vol. abs/1803.03095, 2018. 3

[37] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 8198–8207. 3, 6, 7

[38] A. Andoni, P. Indyk, T. Laarhoven, I. Razenshteyn, and L. Schmidt, "Practical and optimal lsh for angular distance," in *Advances in Neural Information Processing Systems*, 2015, pp. 1225–1233. 3

[39] B. Mandelbrot, "The pareto-levy law and the distribution of income," *International Economic Review*, vol. 1, no. 2, pp. 79–106, 1960. 3

[40] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262. 3

[41] F. Leone, L. Nelson, and R. Nottingham, "The folded normal distribution," *Technometrics*, vol. 3, no. 4, pp. 543–550, 1961. 4

[42] Z. Liu and I. Tsang, "Approximate conditional gradient descent on multi-class classification," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 2301–2307. 4

[43] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511.07122*, 2015. 5

[44] K. Chen, C. C. Loy, S. Gong, and T. Xiang, "Feature mining for localised crowd counting," in *British Machine Vision Conference, BMVC 2012, Surrey, UK, September 3-7, 2012*, 2012, pp. 1–11. 5

[45] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3253–3261. 5

[46] Y. Wang and Y. Zou, "Fast visual object counting via example-based density estimation," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3653–3657. 5

[47] B. Sheng, C. Shen, G. Lin, J. Li, W. Yang, and C. Sun, "Crowd counting via weighted vlad on a dense attribute feature map," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1788–1797, 2018. 5

[48] D. Babu Sam, S. Surya, and V. Babu R, "Switching convolutional neural network for crowd counting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1708–1716. 5, 7

[49] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2547–2554. 6, 7

[50] V. A. Sindagi and V. M. Patel, "Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *Proc AVSS*. IEEE, 2017, pp. 1–6. 7

[51] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Máadeed, N. M. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *ECCV*, 2018. 7

[52] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on Knowledge & Data Engineering*, no. 9, pp. 1263–1284, 2008. 6

[53] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018. 8, 9, 10

[54] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 215–223. 9

[55] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu, "Benchmarking adversarial robustness," 2019. 9

[56] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8571–8580. 9, 10

[57] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," in *International Conference on Learning Representations*, 2018. 9, 10