This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

Consistency-Aware Anchor Pyramid Network for Crowd Localization

Xinyan Liu, Guorong Li, Yuankai Qi, Zhenjun Han, Anton van den Hengel, Nicu Sebe, Ming-Hsuan Yang and Qingming Huang

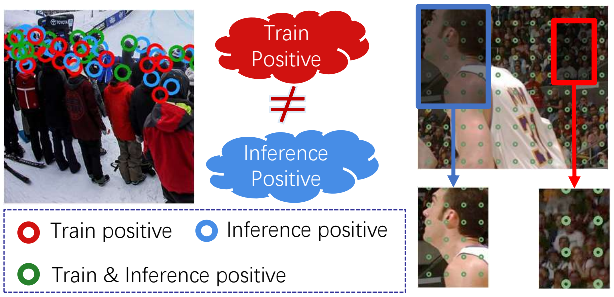
Abstract—Crowd localization aims to predict the positions of humans in images of crowded scenes. While existing methods have made significant progress, two primary challenges remain: (i) a fixed number of evenly distributed anchors can cause excessive or insufficient predictions across regions in an image with varying crowd densities, and (ii) ranking inconsistency of predictions between the testing and training phases leads to the model being sub-optimal in inference. To address these issues, we propose a Consistency-Aware Anchor Pyramid Network (CAAPN) comprising two key components: an Adaptive Anchor Generator (AAG) and a Localizer with Augmented Matching (LAM). The AAG module adaptively generates anchors based on estimated crowd density in local regions to alleviate the anchor deficiency or excess problem. It also considers the spatial distribution prior to heads for better performance. The LAM module is designed to augment the predictions which are used to optimize the neural network during training by introducing an extra set of target candidates and correctly matching them to the ground truth. The proposed method achieves favorable performance against state-of-the-art approaches on five challenging datasets: ShanghaiTech A and B, UCF-QNRF, JHU-CROWD++, and NWPU-Crowd. The source code and trained models will be released at https://github.com/ucasyan/CAAPN

Index Terms—Anchor Pyramid, Augmented Matching, Crowd Localization, Crowd Counting

1 Introduction

The goal of crowd localization is to localize individuals in crowds using point annotations. This problem has received much attention due to a wide range of applications, such as traffic flow analysis [1], medical cell assay [2], and crowd anomaly detection [3]. Despite significant advances that have been made, crowd localization remains challenging partly due to the large variations in density across diverse crowd scenarios.

Existing methods for crowd localization can be broadly categorized into three groups based on their regression targets: detection-based methods, which regress bounding boxes of heads [4]–[8]; point regression, which directly regress point annotations [9], [10]; and heuristic methods, which regress heads in a density map [11], [12] or a segmentation map [13]–[16]. Detection-based methods formulate crowd localization as a typical object detection task and use the center coordinates of the predicted bounding boxes as head locations. The limited number of bounding box annotations [4]–[6] heavily constrains recent advances in detection-based methods. Depth information is used in [7], [8] to estimate head size without bounding box annotations. Heuristic approaches employ various auxiliary maps, such as density maps, segmentation maps, and confidence maps, to capture crowd distribution. These methods require non-differentiable post-processing steps (*e.g.*, finding maxima [11], [17], [18] or finding connected components [13], [14]) to compute head coordinates, making them incapable of being end-to-end trained. On the contrary, point regression methods [9], [10], which also follow the detection paradigm, can directly predict the coordinates of targets. Our work belongs to this category.



(a) (b)

Fig. 1: (a) Illustration of the ranking inconsistency of predictions between the training and testing phases, which may lead to sub-optimal inference performance. (b) Excessive or insufficient numbers of evenly distributed anchors across sparse and dense regions in an image cause performance reduction.

Xinyan Liu, Guorong Li, and Qingming Huang are with the School of Computer Science and Technology, Key Lab of Big Data Mining and Knowledge Management, University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: liuxinyan19@mails.ucas.ac.cn, liguorong@ucas.ac.cn, qmhuang@ucas.ac.cn
Zhenjun Han is with the School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: hanzhj@ucas.ac.cn
Yuankai Qi and Anton van den Hengel are with the Australian Institute for Machine Learning, the University of Adelaide, Adelaide, SA, 5005, Australia, E-mail: qykshr@gmail.com, Anton.vandenHengel@adelaide.edu.au
Nicu Sebe is with the Department of Information Engineering and Computer Science, University of Trento, 38123 Trento, Italy. E-mail: niculae.sebe@unitn.it
Ming-Hsuan Yang is with the University of California at Merced, Merced, CA 95343 USA, and Google, Mountain View, CA 94043 USA. E-mail: mhyang@ucmerced.edu.

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

Despite significant progress in crowd localization, the performance of prevailing point regression methods is limited in two aspects. One limitation is the ranking inconsistency of predictions between the training and inference phases. During inference, the selection of predictions is solely based on classification scores. However, during training, the top-M (M is the number of targets in the image) predictions are selected based on both spatial distance to targets and classification scores. This inconsistency leads the model to be sub-optimized with respect to its testing. We show one example in Figure 1(a), where part of the predictions used for loss computation (denoted as "train positive") are not selected as final results (marked as "inference positive") for inference and thus distract the training process. The other limitation comes with utilizing a fixed number of evenly distributed anchors. An image may contain diverse crowd densities across regions, as shown in Figure 1(b). Using a fixed number of evenly distributed anchors across an image could lead to excessive predictions in regions with sparse targets and inadequate predictions in regions with dense targets, thereby limiting overall performance.

To address these problems, we propose Consistency-Aware Anchor Pyramid Network (CAAPN) for crowd localization, which consists of two main components: an Adaptive Anchor Generator (AAG) and a Localizer with Augmented Matching (LAM). The AAG module is designed to generate anchors according to the estimated density in each local region and spatial distribution prior. Therefore, AAG contains a counting branch, which predicts the number of heads in a region. Existing counting loss (*i.e.*, Mean-Squared Error) is susceptible to inevitable shifts in manual annotations, making the predicted density map less precise to guide anchor distribution. To alleviate this issue, we propose a Cascade Region Loss (CRL) to generate a more precise density map. The distribution prior is gathered from training data in a region-wise manner. The adaptively generated anchors are then fed to the localizer in LAM to make location predictions. As such, the AAG module enables dynamic anchor generation and makes the number and distribution of anchors closer to target as shown in Sec. 3.1. The LAM module, unlike previous methods, selects two groups of top-M predictions according to independent criteria: one group is chosen according to both distance error and classification score similar to existing methods [9], [10]; and the other group is chosen based solely on classification score to keep consistent with the test phase. To effectively utilize it, we assign this group to a specific ground truth set selected according to inverse probability ranking. Ablation study shows that this simple design largely alleviates the ranking inconsistency problem and significantly boosts performance.

The main contributions of this paper are:

- We propose an Adaptive Anchor Generator (AAG) to adaptively generate anchors in each region of an image, which can alleviate the anchor deficiency or excess problem. This module also reduces the computation load in the Hungarian Matching procedure.
- We propose a Localizer with Augmented Matching (LAM) for point regression crowd localization, easing ranking inconsistency between training and testing.
- We propose a cascade regression loss (CRL) to relieve the localization shift error.
- Extensive experiments on five benchmarks, ShanghaiTech A&B, UCF-QNRF, JHU-CROWD++, and NWPU-Crowd, demonstrate the effectiveness of our method compared against several state-of-the-art approaches.

2 Related Work

In this section, we first briefly review existing crowd localization methods, which can be broadly classified into three categories: detection-based, heuristic, and point regression. We then introduce how the inconsistency problem is handled in object detection tasks.

2.1 Existing Crowd Localization Method

Detection-based Methods. Detection-based methods usually require bounding box annotations for training [5], [19]–[21], which is labor-costly in crowd scenes. To overcome this problem, some methods [22]–[24] estimate pseudo bounding boxes based on the distances of neighboring point annotations. However, these pseudo bounding boxes introduce much noise, making the performance of this kind of method far from satisfying. Depth information is useful to generate prior anchors for that it indicates the size of objects [7], [8]. RDNet [7] estimates the bandwidth of the Gaussian kernel for heads in images with depth information. DPDNet [8] proposes a depth-aware anchor to put more anchors in a deeper place, while our method directly puts more anchors in a crowded place based on a preliminary density prediction.

Heuristic Methods. This line of work dominates crowd localization. These methods predict head coordinates with the aid of various intermediate maps. For instance, density maps are utilized in [11], [12], [25]–[27], and its local maxima are viewed as head locations. In a density map, each element denotes the probability of that position being heads and the sum of all elements equals the total count of targets. The quality of density maps plays an essential role in these methods. Haroon *et al.* [26] use cascade adaptive Gaussian kernels to refine the density map to be close to the point map. BL [11] directly models the maximum posterior probability of the ground truth points, thus relieving the need for the blurred Gaussian kernels. In DM-Count [12], optimal transport loss and entropy regularization are adopted to sharpen density maps and facilitate localization. OT-M [28] proposes a general method to iteratively convert the density map to the point map using optimal transport, but the inference phase is time-consuming to solve multiple optimal transport problems. The D2CNet [27] estimates possible head boundaries to generate a probability map with clear margins within heads. In another line of research, segmentation maps (*e.g.*, heads marked with 1, other pixels marked with 0) are leveraged [13]–[15], [29]. These approaches generate head coordinates by finding the center of a connected component in the predicted segmentation map. Their performances suffer from the unavailability of high-quality ground truth segmentation maps from point

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

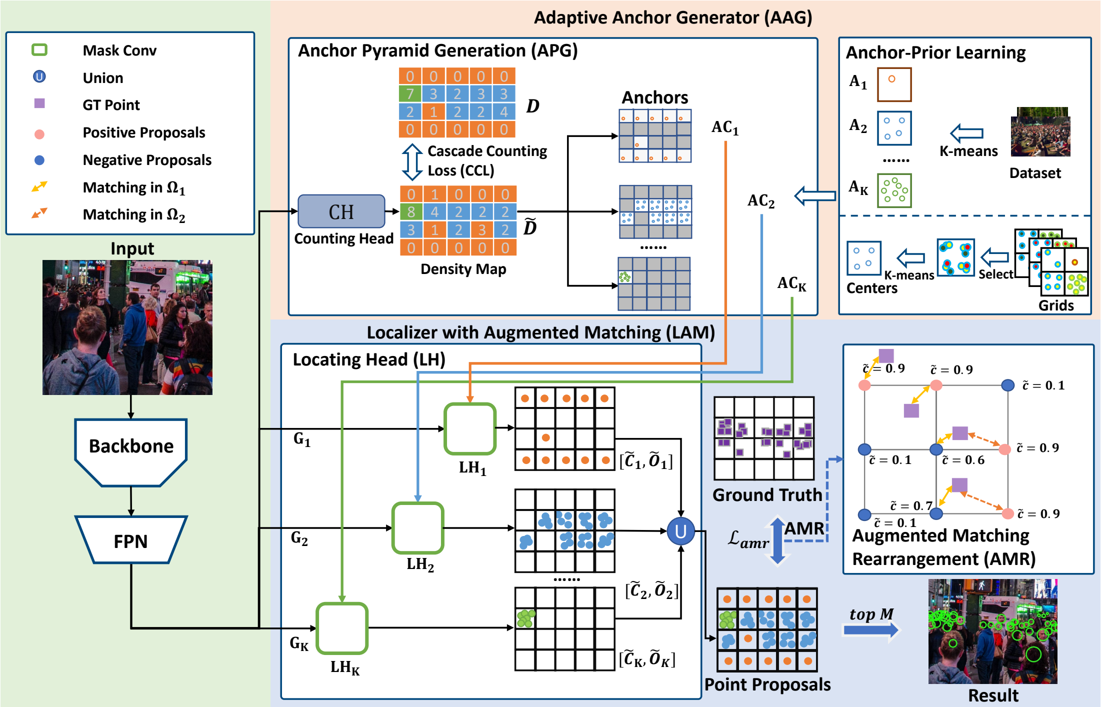


Fig. 2: Main architecture of the proposed method. The input image is evenly divided into grids. Then, the Adaptive Anchor Generator (Sec. 3.1) generates high-quality anchors according to the estimated head number and spatial distribution prior. Next, these anchors are fed to the Localizer with Augmented Matching (Sec. 3.2) module to predict head coordinates. This module is enhanced by introducing a re-matching process of an extra set of target candidates, which alleviates the ranking discrepancy between the training and testing phases.

annotations and the impact of partial occlusion of heads. To make objects more separable in crowd scenes, FIDT [18] proposes to use a focal inverse distance transform rather than a density map. However, it still needs post-processing to obtain head coordinates, which are hand-crafted and cannot be optimized with the main model.

Point Regression Methods. This type of method directly predicts head coordinates. The P2PNet [9] takes in VGG-16 features and uses two branches to simultaneously predict coordinates and classification scores based on a set of evenly distributed anchors. CLTR [10] adopts transformer architecture and applies a set of pre-defined queries to predict head locations. Based on the one-to-one match, a KMO match strategy is proposed in CLTR, which considers the structure of crowds when assigning proposals to targets. Although previous methods perform worse than SoTA heuristic methods, we continue the exploration of this line of research due to its neat design as a whole. Furthermore, with our adaptive pyramid anchor and augmented matching strategy, our method sets new state-of-the-art even compared to heuristic approaches.

2.2 Inconsistency in Object Detection

The aforementioned inconsistency problem also exists in query/anchor-based object detection methods [4], [30]–[32], which often use IoU and classification score together during training but only use classification score during inference. To mitigate this inconsistency problem, the one-to-many label assignment paradigm, which assigns one ground truth object to many queries/anchors, is widely used in object detection methods [33]–[36]. Since more proposals are selected, the probability of the proposals with top-M classification scores being optimized increases.

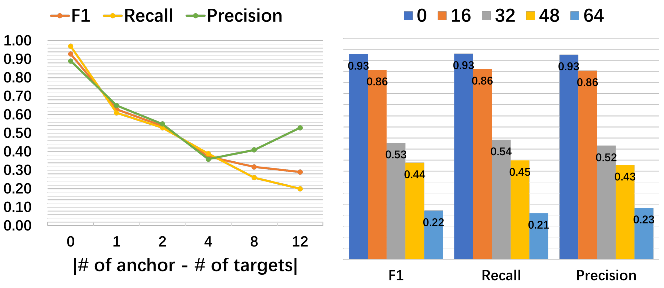
Specifically, FCOS [33] selects pixels within ground truth bounding boxes to predict results. ATSS [34] selects anchors with top-K highest IoU to the ground truth objects. PAA [35] uses an auxiliary branch to predict IoU rather than using ground truth IoU to select anchors. In [36], multiple parallel auxiliary heads are designed, and each head is supervised by a one-to-one label assignment, avoiding NMS (non-maximum suppression) operation. Different from one-to-many label assignment, DDQ [37] first selects proposals with high classification scores, then employs NMS to obtain distinct proposals, which are used for further one-to-one label assignment. However, FCOS [33], ATSS [34], PAA [35], and DDQ [37] require bounding box information to perform NMS to remove extra predictions or proposals. As the bounding box information is unavailable in crowd counting and localization, these methods can not be used in our task.

3 Method

Given an image $I \in \mathbb{R}^{H \times W \times 3}$, the goal of crowd localization is to predict the positions of all heads in the image. As shown in Figure 2, our method is composed of two main components: an Adaptive Anchor Generator (AAG) and a Localizer with Augmented Matching (LAM). The former dynamically generates anchors for each image region; the latter takes anchors as input and predicts head locations. It

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013



(a) (b)

Fig. 3: Performance effect caused by (a) the difference in number between anchors and ground truth; and (b) location offset of anchors to ground truth. Here we give examples at distances of 0, 16, 32, 48, and 64 pixels, respectively.

boosts localization performance by mitigating the inconsistency of prediction selection between training and testing.

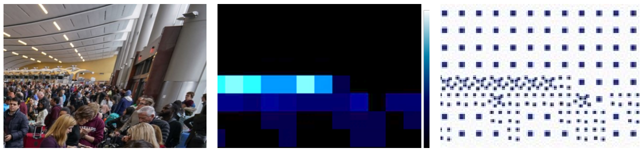
3.1 Adaptive Anchor Generator

Our model predicts locations on the basis of anchors. Unlike previous works which use evenly distributed anchors, we propose to scatter anchors region-wise. This idea is inspired by our observation, as shown in Figure 3. Figure 3(a) presents the performance curve of the SoTA model P2PNet [9] when the number of its anchors differs from the ground truth at different degrees. It shows that the performance becomes worse as the quantity difference increases. In Figure 3(b), we present the performance bars when the same number of anchors are distributed at different distances to targets. It indicates that the performance worsens when anchors get farther away from targets. Thus, we propose to adaptively determine the number of anchors in each region and scatter these anchors according to spatial distribution priors. Below we present the details on how to compute the priors and how to determine the number of anchors accurately.

Distribution Priors. For a fixed-size region, heads are usually distributed at different positions under different head densities. Thus, we would like to gather a series of distribution priors under different densities. Assuming we need priors for K densities $s_1, \cdots, s_K$ where $s_1 < s_2 < \cdots < s_K$, below we detail the computation for a single density s_i as an example. We first partition each training image into regions of size 16×16 pixels. Then, we select regions with the number of heads that fall in $(s_i, s_{i+1}]$. Annotations in these regions are merged into one region according to their local coordinates within 16×16. Next, we perform the K-means [38] clustering algorithm on the merged region to get s_i clusters. The centers of these clusters are the desired distribution prior for density s_i, denoted as $\mathbf{A_i} = \{a_{i,1}, a_{i,2}, ...a_{i,s_i}\}$. As such, we compute K distribution priors for anchor generation.

Anchor Pyramid Generation. Given an image I, we first estimate the number of objects in it by a counting branch, denoted by $\mathbf{CH}$, which is a stack of five convolution layers:

$$\tilde{D} = \mathbf{CH}(\mathbf{F}), \tag{1}$$



(a) Image (b) Density Map (c) Anchors

Fig. 4: Illustration of our anchor pyramid with three levels ($K = 3, s_1 = 1, s_2 = 4, s_3 = 8$).

where $\mathbf{F}$ is the feature extracted by a pre-trained image classification model (*e.g.*, VGG [39], HRNet [40], ConvNeXt [41]), and $\tilde{D}$ is the predicted density map. The learning target D is obtained by pooling the point annotations ($\{0,1\}^{H\times W}$) using *sum* as the pooling operator. The pooling stride is 16, and thus each element of D is the number of heads in a 16×16-pixel region, corresponding to the prior distribution size. The anchor distribution prior for each 16×16 image region is selected according to which density interval its counting prediction falls in:

$$\mathbf{AC}^i_{u,v} = \begin{cases} \mathbf{A_i}, & \text{if } s_{i+1} < \tilde{D}_{u,v} \le s_i \\ \emptyset, & \text{others} \end{cases} \tag{2}$$

where i indicates the index of distribution prior ranging from 1 to K. The resulting $\mathbf{AC}^i$ will be used for location prediction, detailed in the next section. Figure 4 provides an example of $\mathbf{AC}^i$. As $\mathbf{AC}^i$ denotes anchor sets of a specific densities used for image I, we refer to $\{\mathbf{AC}^1, \mathbf{AC}^2, \cdots, \mathbf{AC}^K\}$ as the anchor pyramid. It shows that guided by the predicted density map, the number and spatial distribution of the generated anchor pyramid are more consistent with that of the crowds.

We note that Eq. (2) shows that the quality of $\mathbf{AC}^i$ depends on the precision of $\tilde{D}$. However, existing loss functions for training a counting network, e.g., Mean-Squared Error (MSE) and Mean-Absolute Error (MAE), are susceptible to inevitable shifts in manual annotations, which might lead to $\tilde{D}$ being less precise. To alleviate this effect, we propose a new loss based on multiple resolutions, denoted as Cascade Counting Loss (CCL), where annotation shifts in higher resolution can be corrected in lower resolutions. Below, we give more details about how to calculate CCL.

We first partition the input image into non-overlapping regions under multi-resolutions $\frac{H}{2^i} \times \frac{W}{2^i}$, where $i \in \{0, 1, 2, \cdots, N_r\}$ and $N_r = \log_2(\min(\frac{H}{16}, \frac{W}{16}))$. These partitions result in sets of regions $R_0, R_1, \cdots, R_{N_r}$, where $R_i = \{r_{1,i}, r_{2,i}, \cdots, r_{2^i \times 2^i, i}\}$ with region resolution $\frac{H}{2^i} \times \frac{W}{2^i}$. Next, we define Δr as the absolute residual of head numbers between D and $\tilde{D}$ on region r:

$$\Delta r = \left| \sum_{(u,v)\in r} D_{u,v} - \sum_{(u,v)\in r} \tilde{D}_{u,v} \right|. \tag{3}$$

Then, we design a CasCade Region Loss (CRL) on those multiple resolutions:

$$\mathcal{L}_{crl} = \sum_{r\in R_0} \frac{1}{Z_0} \Delta r + \sum_{i=1}^{N_r} \sum_{r\in R_i} \frac{1}{Z_i} \underbrace{\frac{\mathrm{e}^{\Delta \hat{r}}}{\mathrm{e}^{\sum_{\hat{r}\in R_{i-1}} \Delta \hat{r}}}}_{\text{re-weight term}} \Delta r, \tag{4}$$

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

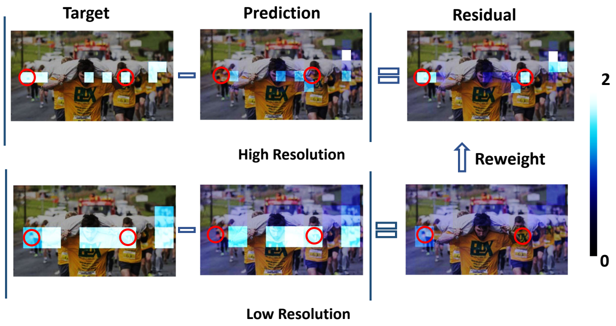


Fig. 5: Illustration of our re-weighting strategy in CRL. At the lower resolution, errors instigated by localization shifts are disregarded, while errors resulting from inadequate or superfluous predictions are retained. Consequently, we utilize lower resolution residuals as weights to direct the model's focus more towards the latter category of errors.

where $\hat{r}$ represents a region in R_{i-1} that covers r; $Z_i = \frac{2^i \times 2^i}{H \times W}$ and $\frac{1}{Z_i}$ denotes the number of elements in r. The re-weight term is designed based on an observation that a counting error caused by shifts in a small region might be corrected in a larger region that covers it. Figure 5 shows an example, where the counting error is 3 in the high resolution (Figure 5 left panel). In contrast, the error reduces to zero in its corresponding region at a lower resolution (Figure 5 middle panel). Thus, we use the normalized errors at a lower resolution as the weights for the current resolution to reduce its effect if it has errors.

Considering that the final output of the counting branch in an inference phase is the integer closest to the predicted decimal, we propose the following Integer Loss (IL) to reflect this fluctuation:

$$\mathcal{L}_{il} = \sum |\lfloor \tilde{D}_{u,v} \rceil - \tilde{D}_{u,v})|, \tag{5}$$

where $\lfloor \tilde{D}_{u,v} \rceil$ is the rounded number of $\tilde{D}_{u,v}$. The final Cascade Counting Loss (CCL) is:

$$\mathcal{L}_{ccl} = \mathcal{L}_{crl} + \mathcal{L}_{il}. \tag{6}$$

3.2 Localizer with Augmented Matching

The localizer takes image features and anchors as inputs. For each anchor, it estimates a head location and the corresponding classification probability of being head. These estimations are usually termed "proposals". Because the number of anchors is much larger than the ground truth, only a small part of the proposals are selected as final predictions to perform the Hungarian Match and then compute the loss to optimize the neural network.

A long-standing problem of existing methods is the inconsistency in selecting final predictions between the training and testing phases. In the training phase, existing methods select the top-M proposals based on the classification probability and the Euclidean distance to ground truth. However, only the probability is used for selection during testing. Such inconsistency may lead to the localizer not being optimized in line with how it is used during testing, limiting its performance. We propose an augmented matching strategy to mitigate this issue. In the following, we describe our localizer module and present our new augmented matching (AM) strategy.

Locating head. We utilize a locating branch with K heads to predict head coordinates, denoted as $\mathbf{LH}_i(\cdot)$. Using the common practice in object detection methods [19], [34], [42], we first extract features of K different scales (denoted as $\mathbf{G}_1, \mathbf{G}_2, \cdots, \mathbf{G}_K$) using FPN [43] from the $1^{th}, 2^{th}, \cdots, K^{th}$ pyramid levels. We then feed $\mathbf{LH}_i(\cdot)$ with $\mathbf{G}_i$ and $\mathbf{AC}^i$ to generate head proposals:

$$[\tilde{C}_i, \tilde{O}_i] = \mathbf{LH}_i(\mathbf{G}_i, \mathbf{AC}^i), \tag{7}$$

where $\tilde{C}_i \in \mathbb{R}^{\tilde{M}_i}$ and $\tilde{O}_i \in \mathbb{R}^{\tilde{M}_i \times 2}$ are the binary classification probabilities and coordinates of the predicted points, respectively; and $\tilde{M}_i$ is the number of proposals, which depends on the number of anchors fed to $\mathbf{LH}_i$. We denote all proposals from K locating heads as $\tilde{P} = \cup_{i=1}^{K}\{\tilde{C}_i, \tilde{O}_i\} = \{\tilde{p}_1, \tilde{p}_2, \cdots, \tilde{p}_{\tilde{M}}\}$, where $\tilde{p}_i = (\tilde{x}_i, \tilde{y}_i, \tilde{c}_i)$ and $\tilde{c}_i$ is the classification probability. $\tilde{M} = \sum_{i=1}^{K} \tilde{M}_i$ is the number of all proposals in $\tilde{P}$.

Augmented Matching. Our augmented matching strategy consists of two steps to select final predictions from $\tilde{P}$ for Hungarian Match during training. First, we select M predictions according to both spatial distance and object probability as existing methods [9], [10], where M is the number of ground truth heads in the image. Specifically, we exploit the Hungarian Match Algorithm [44] to get the optimal matching Ω_1 between the ground truth point annotation set P and the prediction set $\tilde{P}$:

$$\Omega_1 = \arg\min_{\Omega} \mathcal{L}_{loc}(P, \tilde{P}, \Omega), \tag{8}$$

where $\mathcal{L}_{loc}(P, \tilde{P}, \Omega)$ is the cost function used to evaluate a one-to-one matching Ω. The optimal matching Ω_1 leads to the smallest cost; $\mathcal{L}_{loc}(P, \tilde{P}, \Omega)$ is defined as:

$$\mathcal{L}_{loc}(P, \tilde{P}, \Omega) = \sum_{\tilde{p}_i \in \tilde{P}} \mathcal{L}_{cls}(\tilde{p}_i) + \mathcal{L}_{dist}(\tilde{p}_i, p_i), \tag{9}$$

where $p_i = \Omega(\tilde{p}_i)$ represents the matched ground truth of $\tilde{p}_i$ under Ω; $\mathcal{L}_{cls}(\cdot)$ is the classification focal loss [45], which can alleviate the imbalance between foreground and background samples; and $\mathcal{L}_{dist}(\cdot, \cdot)$ is the L2 distance. We denote the set of matched M predictions determined by Ω_1 as S_1.

Next, we re-select M predictions according to classification probability $\tilde{p}$, which aligns with the criteria utilized during testing. The M proposals with top classification probabilities, denoted as S_2, are selected. Then, we need to match these predictions to ground truth annotations. We notice that part of these proposals also appears in S_1, which has already been assigned to ground truth. Thus, we need to assign the remaining ones, denoted as $S' = S_2 - (S_2 \cap S_1)$ to a ground truth set. Assuming there are M' predictions in S', we firstly select M' predictions with the lowest M' classification probabilities from S_1. As these M' predictions have lower classification scores, they are not likely to be chosen as final predictions during the inference phase. We re-assign their matched ground truth annotations (denoted as P') to S'. We name this strategy as Inverse Probability (IP).

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

TABLE 1: Comparison of localization performance against State-of-the-Art Methods on the ShanghaiTech A &B (STA, STB) datasets. The main metric is F1 under $\sigma = 8$. The best and second best results are highlighted in red and blue, respectively.

Methods	Features	STA						STB					
		$\sigma = 4$			$\sigma = 8$			$\sigma = 4$			$\sigma = 8$		
		F1	P	R	F1	P	R	F1	P	R	F1	P	R
LOWB [25]	UNet	25.9	34.9	20.7	53.9	67.7	44.8	-	-	-	-	-	-
LSC-CNN [23]	VGG-16	32.6	33.4	31.9	62.4	63.9	61.0	29.5	29.7	29.2	57.0	57.5	56.7
TopoCount [13]	VGG-16	41.1	41.7	40.6	73.6	74.6	72.7	63.2	63.4	63.1	82.0	82.3	81.8
CLTR [10]	Transformer	43.2	43.6	42.7	74.2	73.5	74.9	-	-	-	-	-	-
FIDT [18]	HRNet-W48	58.6	58.2	59.1	77.6	78.2	77.0	64.7	64.9	64.5	83.5	83.9	83.2
CAAPN (Ours)	VGG-16	57.5	56.3	58.7	78.0	78.8	77.2	65.3	65.4	65.2	83.3	82.4	84.2
CAAPN (Ours)	ConvNeXt-S	60.3	58.9	61.7	78.3	79.1	77.5	65.7	66.4	65.0	84.9	84.7	85.0
CAAPN (Ours)	HRNet-W48	59.9	60.0	59.8	78.5	78.3	78.7	65.3	64.9	65.6	83.8	83.4	84.2

TABLE 2: Localization performance on the UCF-QNRF dataset. The main metric is F1. The best and second best results are highlighted in red and blue, respectively.

Methods	Features	F1	P	R
LOWB [25]	UNet	60.05	75.46	49.87
LSC-CNN [23]	VGG-16	74.06	74.62	73.50
TopoCount [13]	VGG-16	80.34	81.77	78.96
CLTR [10]	Transformer	80.97	82.22	79.75
FIDT [18]	HRNet-W48	82.23	84.49	80.10
CAAPN (Ours)	VGG-16	81.61	84.06	79.30
CAAPN (Ours)	ConvNeXt-S	82.58	83.78	81.41
CAAPN (Ours)	HRNet-W48	83.26	86.92	79.89

We then match S' to ground truth annotations in P' based on only the spatial distance. Using the Hungarian Match Algorithm, the optimal match Ω_2 between S' and P' can be achieved by

$$\Omega_2 = \arg\min_{\Omega} \sum_{\tilde{p}_i \in S'} \mathcal{L}_{dist}(p_i, \tilde{p}_i). \tag{10}$$

where Ω is a potential matching and $p_i = \Omega(\tilde{p}_i)$.

As our matching strategy expands the conventional matching, we name it Augmented Matching. Based on it, our localization loss is defined as:

$$\mathcal{L}_{am}(P, \tilde{P}) = \mathcal{L}_{loc}(P, \tilde{P}, \Omega_1) + \sum_{\tilde{p}_j \in S'} \mathcal{L}_{dist}(p_j, \tilde{p}_j), \tag{11}$$

where $p_j = \Omega_2(\tilde{p})$.

Discussion. The above-mentioned IP strategy is not the only option for matching. For example, we can: (i) find the matching to S' from all ground truth annotations according to Eq. (9) or (ii) select M' ground truth with the highest matching cost under the optimal matching Ω_1. We term alternative (i) as direct rearrangement (DR) and alternative (ii) as high-cost rearrangement (HCR). As shown in our ablation studies (Section 4.3), all these augmenting strategies improve the localization performance compared to no augmentation but our inverse-probability approach performs best.

4 Experimental Results

We first present the implementation details and briefly introduce the five evaluation benchmarks (ShanghaiTech A and B [54], UCF-QNRF [26], JHU-CROWD++ [55], and NWPU-Crowd [56]) as well as the corresponding evaluation metrics. Then, we evaluate our method against several state-of-the-art methods. In addition, we provide a thorough ablation study of the proposed method.

TABLE 3: Localization performance on the JHU-CROWD++ dataset. The main metric is F1 under $\sigma = 8$. The best and second best results are highlighted in red and blue, respectively.

Methods	Features	$\sigma = 4$	$\sigma = 8$
		F1 / P / R	F1 / P / R
TopoCount [21]	VGG-16	30.1/31.5/28.8	60.8/63.6/58.3
FIDT [18]	HRNet-W48	38.8/38.9/38.7	62.4/62.4/62.5
CAAPN (Ours)	VGG-16	31.2/31.4/31.1	62.2/66.0/58.8
CAAPN (Ours)	ConvNeXt-S	40.2/39.9/40.5	64.2/63.4/65.0
CAAPN (Ours)	HRNet-W48	40.4/40.1/40.8	65.6/65.4/65.8

TABLE 4: Localization performance on the NWPU-Crowd dataset. The main metric is F1 under σ_l. The best and second best results are highlighted in red and blue, respectively.

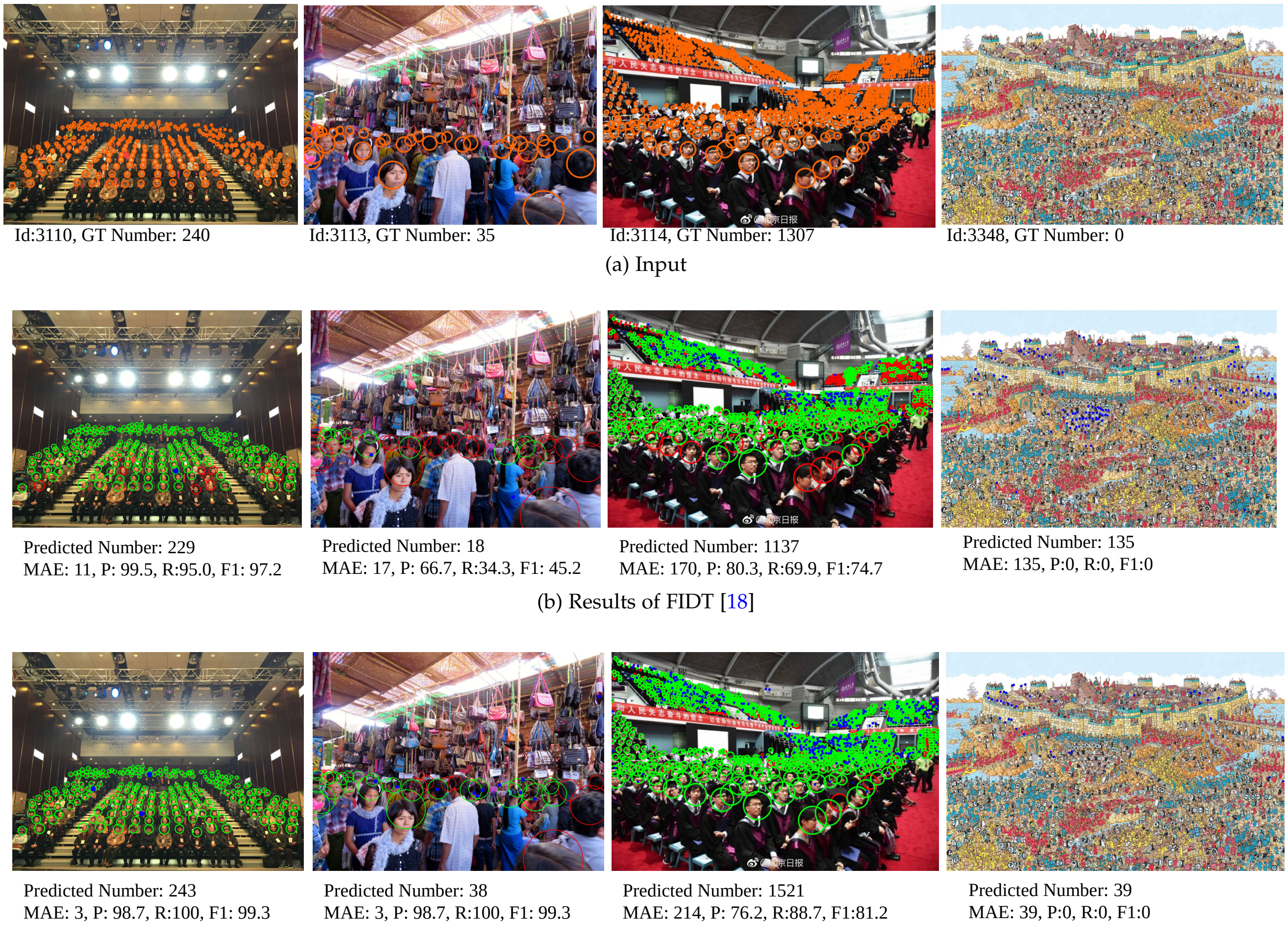
Methods	Features	σ_l	σ_s
		F1 / P / R	F1 / P / R
TinyFaces [21]	ResNet-101	56.7/52.9/61.1	52.6/49.1/56.6
TopoCount [13]	VGG-16	69.1/69.5/68.7	60.1/60.5/59.8
RAZLoc [46]	VGG-16	59.8/66.6/54.3	51.7/57.6/47.0
AutoScale [47]	VGG-16	62.0/67.3/57.4	54.4/59.1/50.4
P2PNet [9]	VGG-16	71.2/72.9/69.5	-/-/-
IIM [14]	HRNet-W48	76.0/82.9/70.2	71.3/77.7/65.8
FIDT [18]	HRNet-W48	75.5/79.7/71.7	70.5/74.4/66.9
DCST [48]	DCST	77.5/82.2/73.4	72.5/76.9/68.6
GMS [49]	HRNet-W48	78.1/79.8/76.5	-/-/-
CAAPN (Ours)	VGG-16	76.5/79.6/73.7	70.4/73.2/67.9
CAAPN (Ours)	ConvNeXt-S	77.8/81.3/74.5	71.5/74.7/68.5
CAAPN (Ours)	HRNet-W48	78.6/80.4/76.8	72.7/74.3/71.1

4.1 Experimental Setups

Implementation Details. During training, the image size is padded to a size that is an integer multiple of 64. Similar to P2PNet [9], for the JHU-Crowd++ and NWPU-Crowd, we limit the longest edge within 1920 and keep the original aspect ratio. We set the number of locating branches $K = 3$.

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013



Id:3110, GT Number: 240
Id:3113, GT Number: 35
Id:3114, GT Number: 1307
Id:3348, GT Number: 0

(a) Input

Predicted Number: 229
MAE: 11, P: 99.5, R:95.0, F1: 97.2

Predicted Number: 18
MAE: 17, P: 66.7, R:34.3, F1: 45.2

Predicted Number: 1137
MAE: 170, P: 80.3, R:69.9, F1:74.7

Predicted Number: 135
MAE: 135, P:0, R:0, F1:0

(b) Results of FIDT [18]

Predicted Number: 243
MAE: 3, P: 98.7, R:100, F1: 99.3

Predicted Number: 38
MAE: 3, P: 98.7, R:100, F1: 99.3

Predicted Number: 1521
MAE: 214, P: 76.2, R:88.7, F1:81.2

Predicted Number: 39
MAE: 39, P:0, R:0, F1:0

(c) Results of our CAAPN

Fig. 6: Visualization of results obtained by FIDT and our CAAPN on NWPU-Crowd validation set. The predicted TP, FN, and FP are denoted as green, blue, and red, respectively.

TABLE 5: Localization performance in terms of recall (%) on the NWPU-Crowd dataset under different object sizes (in pixels). The best and second best results are highlighted in red and blue, respectively.

Methods	Features	$[10^0, 10^1]$	$(10^1, 10^2]$	$(10^2, 10^3]$	$(10^3, 10^4]$	$(10^4, 10^5]$	$(10^5, +\infty)$	Avg
		σ_s, σ_l	σ_s, σ_l	σ_s, σ_l	σ_s, σ_l	σ_s, σ_l	σ_s, σ_l	σ_s, σ_l
RAZLoc [46]	VGG-16	5, 5	21, 28	43, 52	75, 80	60, 64	16, 25	37, 42
TopoCount [13]	VGG-16	5, 6	27, 39	62, 72	82, 86	83, 87	82, 90	56, 63
IIM [14]	HRNet-W48	10, 12	38, 45	69, 73	80, 83	62, 64	11, 17	45, 49
FIDTM [45]	HRNet-W48	19, 22	59, 67	71, 76	68, 72	34, 37	6, 10	43, 48
DCST [48]	DCST	12, 14	44, 51	71, 75	81, 84	77, 81	51, 58	56, 61
CAAPN (Ours)	VGG-16	12, 14	46, 56	68, 74	83, 86	82, 86	66, 73	59, 65
CAAPN (Ours)	ConvNext-S	11, 13	44, 53	69, 75	84, 87	83, 87	68, 76	60, 65
CAAPN (Ours)	HRNet-W48	14, 16	49, 59	73, 78	83, 87	79, 84	55, 64	59, 65

s_1, s_2 and s_3 are set to 1, 4, and 8 for all datasets except UCF-QNRF. For the UCF-QNRF dataset, s_1, s_2 and s_3 are set to 1, 8, and 16 as in P2PNet [9]. We use feature maps from different stages with the same size (stages 3, 4, and 5 for VGG-16 and stages 1, 2, and 3 for ConvNeXt-S). When using HRNet-W48 to extract features, we select feature maps at stage four, as in IIM [14] and FIDT [18]. All these feature extraction models are pre-trained on ImageNet, taken from pytorch image model (Timm [57]). We adopt AdamW as the optimizer, and the learning rate is set to 1e-4 with a cosine scheduler. The experiments are conducted on one RTX 3090 GPU card (24 GB GPU Memory).

Datasets. We use ShanghaiTech A and B, UCF-QNRF, JHU-CROWD++, and NWPU-Crowd datasets to evaluate our method. The ShanghaiTech A dataset contains web images with high crowd densities, while the ShanghaiTech B dataset includes street images with relatively sparse crowds. The UCF-QNRF dataset presents a more challenging scenario with high-resolution images and a wide range of human counts, ranging from 49 to 12,865 across 1,525 images. The JHU-CROWD++ dataset covers diverse scenarios and environmental conditions, consisting of 4,250 images with

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

TABLE 6: Comparison of counting performance against state-of-the-art methods. The main metric is MAE. The best and second best results are highlighted in red and blue, respectively.

Methods	Output coordinates	Features	NWPU-Crowd		JHU-CROWD++		UCF-QNRF		STA		STB	
			MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
CSRNet [50]	no	VGG-16	121.3	387.8	85.9	309.2	-	-	68.2	115.0	10.6	16.0
MBTTBF [51]	no	VGG-16	-	-	81.8	299.1	97.5	165.2	60.2	94.1	8.0	15.5
BL [11]	no	VGG-19	105.4	454.2	75.0	299.9	88.7	154.8	62.8	101.8	7.7	12.7
AMSNet [52]	no	AMSNet	-	-	-	-	101.8	163.2	56.7	93.4	6.7	10.2
NoisyCC [53]	no	VGG-19	102.6	398.4	67.7	258.5	85.8	150.6	61.9	99.6	7.4	11.3
DM-Count [12]	no	VGG-19	88.4	388.6	-	-	85.6	148.3	59.7	95.7	7.4	11.8
RAZ [46]	yes	VGG-16	-	-	-	-	118.0	198.0	71.6	120.1	9.9	15.6
LSC-CNN [23]	yes	VGG-16	-	-	112.7	454.4	120.5	218.2	66.4	117.0	8.1	12.7
AutoScale [47]	yes	VGG-16	94.2	388.2	85.6	356.1	104.4	174.2	65.8	112.1	8.6	13.9
P2PNet [9]	yes	VGG-16	72.6	331.6	-	-	85.3	154.5	52.7	85.1	6.2	9.9
TopoCount [13]	yes	VGG-16	-	-	60.9	267.4	89.0	159.0	61.2	104.6	7.8	13.7
FIDT [18]	yes	HRNet-W48	86.0	312.5	66.6	253.6	89.0	153.5	57.0	103.4	6.9	11.8
CLTR [10]	yes	Transformer	-	-	59.5	240.6	85.8	141.3	56.9	95.2	6.5	10.6
GMS [49]	yes	HRNet-W48	-	-	-	-	104	197.4	68.8	138.6	16.0	33.5
CAAPN (Ours)	yes	VGG-16	71.5	289.7	58.3	236.6	83.9	144.3	54.4	97.3	5.8	9.8
CAAPN (Ours)	yes	HRNet-W48	79.7	341.2	59.9	242.6	85.3	149.3	54.7	99.8	6.1	10.4
CAAPN (Ours)	yes	ConvNeXt-S	76.2	332.0	60.8	250.9	87.5	138.5	54.6	100.5	5.9	10.6

crowd counts ranging from 0 to 7,286. Finally, the NWPU-Crowd dataset provides 5,109 images with a wide range of human counts (including 351 images without humans).

Evaluation Metrics. For the counting performance, we adopt the widely used Mean Absolute Error (MAE) and Mean Squared Error (MSE) as metrics. For the localization performance, we use Precision, Recall, and F1-measure (P, R, F1 for short) for evaluation. Following the setting in FIDT [18], different datasets use different criteria for judging a prediction as true positive. Specifically, datasets ShanghaiTech A and B and JHU-CROWD++ datasets adopt two distance thresholds: 4 pixels and 8 pixels. The UCF-QNRF dataset takes a series of thresholds from 1 to 100 with a step size of 1. It computes the average recall, precision, and F1 as the final performance metric. The NWPU-Crowd dataset utilizes thresholds related to the size of targets. For strict localization setting, the threshold σ_s^i for ground truth point i is set by $\sigma_s^i = 0.5 \times \min(h_i, w_i)$. For a relatively loose localization setting, the threshold is set to $\sigma_l^i = 0.5 \times \sqrt{h_i^2 + w_i^2}$.

4.2 Comparisons to the State-of-the-art Methods

We note that existing methods utilize different image features. For fair comparisons, we evaluate our method using three different features obtained via VGG-16, HRNet-W48, and ConvNeXt-S, respectively.

ShanghaiTech A&B. The datasets STA and STB focus on dense and sparse scenes, respectively. As shown in Table 1, with VGG-16 [39], our CAAPN significantly outperforms the methods using the same features (eg. 78.0 of CAAPN vs 73.6 of TopoCount using $\sigma = 8$) and CLTR which uses the advanced Transformer [30] with nearly three times of parameters as ours (43M v.s. 15M). When adopting the same feature as FIDT, our CAAPN achieves better F1 than FIDT on these two datasets and achieves the best results on STA: 77.6 v.s. 78.5 on STA, and 83.5 VS 83.8 on STB. When using ConvNeXt-S to extract image features, our CAAPN achieves the best results on STB.

UCF-QNRF. This dataset consists of high-resolution images and congested crowds. As shown in Table 2, our method achieves not only the best F1 score but also significantly increases the precision (2.43 over the previous best FIDT) using the same features.

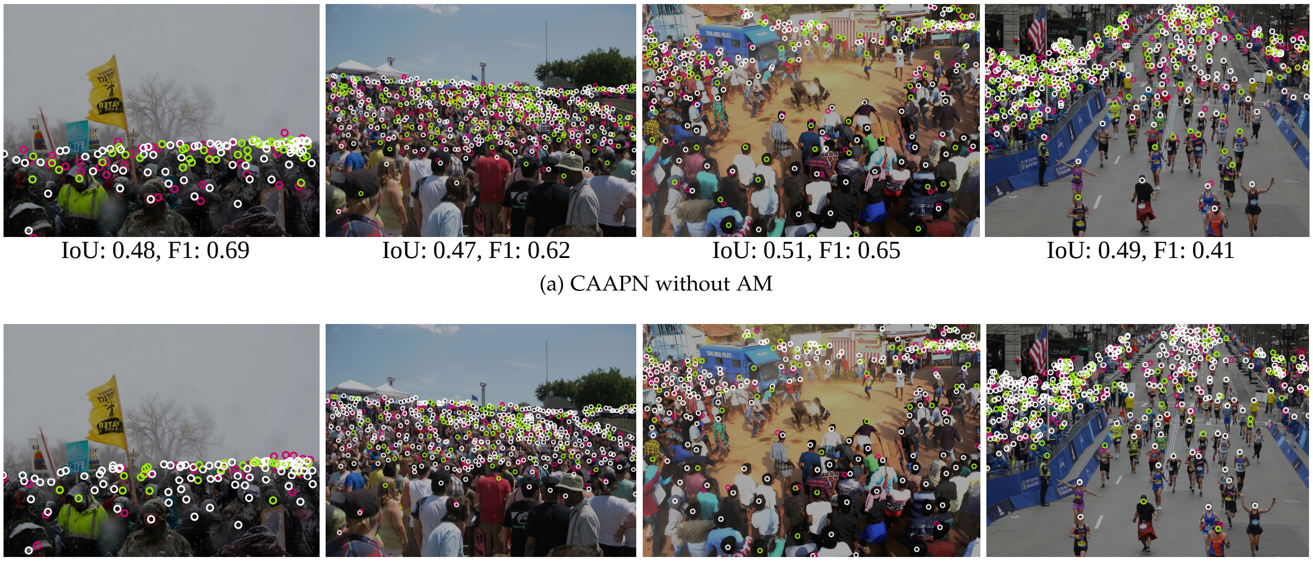
JHU-CROWD++. This dataset has a rich diversity in crowd density and scenes. For this dataset, we only find publicly available results of TopoCount and FIDT. The results are presented in Table 3. Using the same feature extractor (VGG-16) as TopoCount, our CAAPN achieves better F1 on both $\sigma = 4$ and $\sigma = 8$ settings, increasing F1 by 3.7% and 2.3%, respectively. Compared to FIDT, our CAAPN also achieves higher performance: ∼2% on all metrics under the setting $\sigma = 4$ and ∼3% under the setting $\sigma = 8$. These results indicate our method is more robust to density variations and scenario changes.

NWPU-Crowd. As shown in Table 4, our method achieves the highest F1 and recall scores under both σ_l and σ_s settings on the test split. Our CAAPN with HRNet-W48 pushes the boundary of F1/R to 78.6/76.8 under the setting σ_l, and to 72.7/71.1 under the setting σ_s. In Table 5, we provide detailed recall scores of objects with different sizes. Our CAAPN with ConvNeXt-S and HRNet-W48 achieves the best and second-best average recall across all object sizes, demonstrating efficacy in handling various object scales. This is because object size is often negatively correlated with object density and CAAPN performs well in both crowd and sparse regions, which can be attributed to the AAG's ability to generate anchors with various object densities adaptively.

In Figure 6, we visualize the results on different target densities. For the medium crowded image Id 3110 (level 2

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013



IoU: 0.48, F1: 0.69 IoU: 0.47, F1: 0.62 IoU: 0.51, F1: 0.65 IoU: 0.49, F1: 0.41

(a) CAAPN without AM

IoU: 0.62, F1: 0.75 IoU: 0.65, F1: 0.70 IoU: 0.67, F1: 0.73 IoU: 0.69, F1: 0.64

(b) CAAPN

Fig. 7: Visulization of the effect of AM, which improves the IoU of proposals selected according to criteria used in training and testing. White circles denote overlapped proposals. Pink ones denote proposals only selected by training criteria. Chartreuse circles denote proposals only selected by testing criteria.

in NWPU-Crowd density label), our CAAPN finds almost all the targets with only 3 incorrect predictions. In this image, most people missed by FIDT are in the front rows and of relatively sparse density. In contrast, our CAAPN can find all these points thanks to the AAG module. For the sparse crowded image Id 3113, which is of various scales (NWPU-Crowd density label 1) in a complex market scene, our method outperforms FIDT by a significant margin on both precision and recall. We attribute this to the region-wise anchor generation and point proposal rearrangement strategy. The image in the third column is not only congested (NWPU-Crowd density label 3) but also low resolution. The density of crowds exceeds the upper bound that FIDT can handle. With our AAG, CAAPN can generate denser anchors in congested regions and thus handle this challenging scenario well. Finally, for the rightmost image, where there are no visible persons, our method still performs well.

Counting Performance. Although this work focuses on crowd localization, we also provide the counting performance for comprehensive evaluation. The results are presented in Table 6. Our CAAPN achieves the best performance on four out of five benchmarks in terms of the main metric MAE and ranks second on the dataset STA, slightly behind P2PNet.

4.3 Ablation Studies

In this section, we thoroughly evaluate the effectiveness of the key components of our method: Augmented Matching Rearrangement (AM) and Anchor Pyramid Generation (APG). We also evaluate the effectiveness of the proposed Cascade Counting Loss (CCL) and Anchor-Prior Learning. All the experiments are conducted on the JHU-CRWOD++ dataset with features extracted by ConvNeXt-S unless specified otherwise.

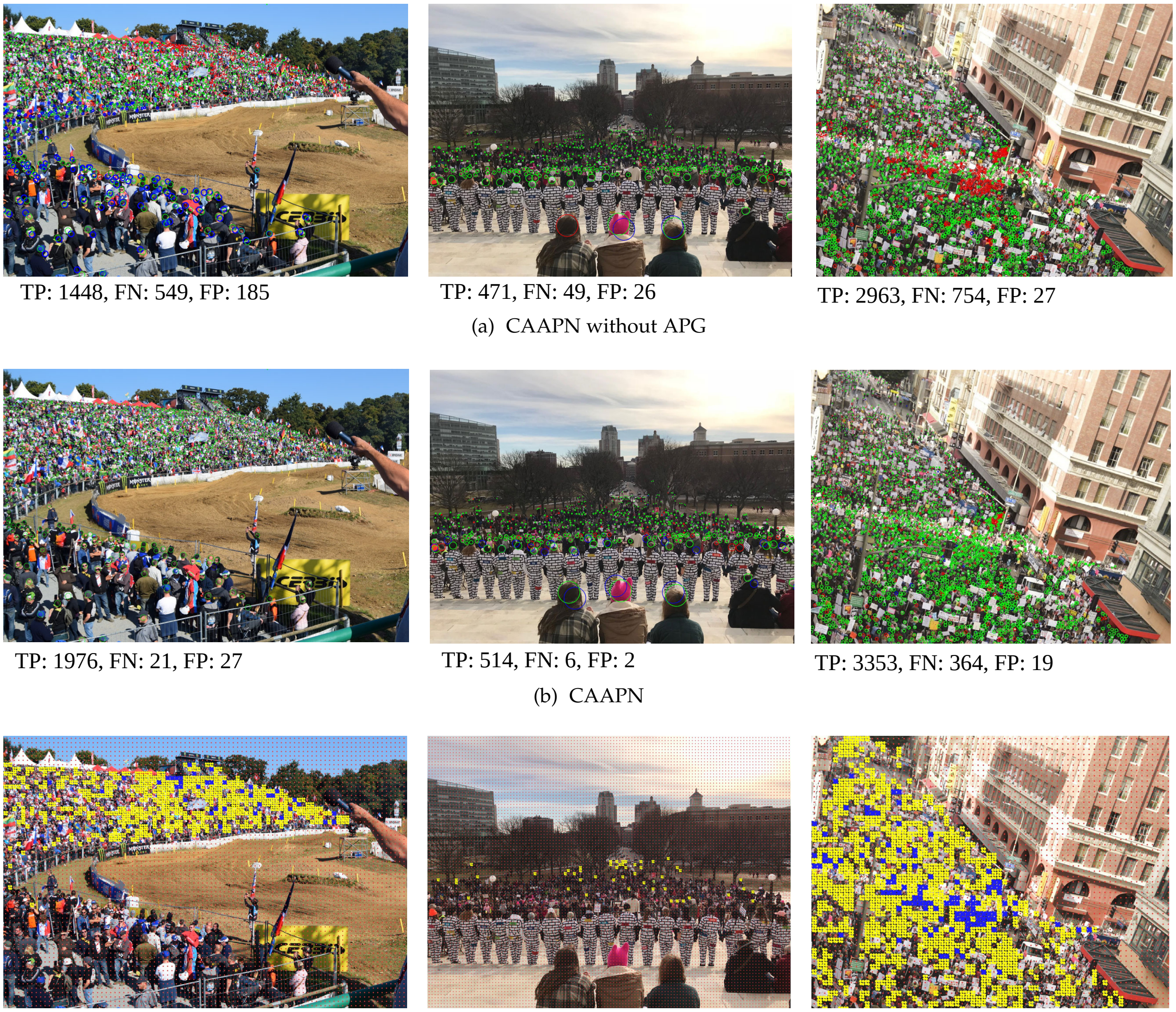
Effectiveness of AM. To evaluate the effectiveness of AM, we remove it and report the results in Table 7. The results show that our AM improves the F1 score (the main metric) by 1.2, 0.7, 0.75, 0.9, and 0.4 on five datasets, respectively. AM strategy is more effective in improving precision for all datasets except NWPU-Crowd. The main reason is the Anchor Redundancy (AR) is lowest on NWPU-Crowd. AR is defined as $AR = \frac{1}{T}\sum_{t=1}^{T}\frac{\tilde{M}_t - M_t}{M_t}$, where T is the number of images in the dataset, M_t and $\tilde{M}_t$ denote the ground truth crowd count and the number of anchors. The AM strategy is designed to reassign anchors to the ground truth. The lowest anchor redundancy on the NWPU-Crowd dataset makes the inconsistency problem less severe and might not release the full potential of our AM. Therefore, the performance gains of the proposed CAAPN are not as significant as on other datasets.

Figure 7 shows some results with proposals only selected to optimize the model during training (red color), proposals only selected for inference (blue color), and selected for both of these two phases (green color). Our augmented strategy makes the selected predicted more consistent and thus improves the performance.

We explore multiple matching strategies for the extra introduced predictions S', including DR, HCR, and IP. With the DR strategy, S' is directly matched to all ground truth annotations. With the HCR strategy, S' is matched to ground truth annotations corresponding to top-M' cost defined by Eq. (9). IP denotes our Inverse Probability ranking strategy. The results are presented in Table 9. It shows that not using the extra matching leads to an obvious performance drop. Compared to the proposed two alternative strategies, *i.e.*, DR (direct rearrangement) and HCR (high-cost rearrangement), the proposed IP strategy shows advantages across all metrics (F1, P, R) and both σs. Especially on $\sigma = 8$, the F1 improvement of IP is 0.9, which is three times higher than

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013



TP: 1448, FN: 549, FP: 185

TP: 471, FN: 49, FP: 26

TP: 2963, FN: 754, FP: 27

(a) CAAPN without APG

TP: 1976, FN: 21, FP: 27

TP: 514, FN: 6, FP: 2

TP: 3353, FN: 364, FP: 19

(b) CAAPN

(c) Anchors generated by our CAAPN

Fig. 8: Qualitative results of our CAAPN and CAAPN without APG. Green, blue, and red circles denote truth positive, false negative, and false positive, respectively. Compared to our CAAPN, CAAPN without APG cannot well fit all density levels. In (c), red, yellow, and blue points denote anchors in $\mathbf{AC}_1$, $\mathbf{AC}_2$, and $\mathbf{AC}_3$, respectively.

TABLE 7: Ablation studies on five datasets.

Dataset	Proposed			w/o AM			w/o APG		
	F1	P	R	F1	P	R	F1	P	R
STA	78.3	79.1	77.5	77.1($\downarrow$1.2)	77.0($\downarrow$2.1)	77.2($\downarrow$0.3)	77.2($\downarrow$1.1)	78.4($\downarrow$0.7)	76.1($\downarrow$1.4)
STB	84.9	84.7	85.0	84.2($\downarrow$0.7)	83.8($\downarrow$0.9)	84.6($\downarrow$0.4)	84.0($\downarrow$0.9)	83.9($\downarrow$0.8)	84.1($\downarrow$0.9)
UCF-QNRF	82.58	83.78	81.41	81.83($\downarrow$0.75)	82.51($\downarrow$1.27)	81.16($\downarrow$0.25)	81.55($\downarrow$1.03)	83.01($\downarrow$0.77)	80.14($\downarrow$1.27)
JHU-Crowd++	65.6	65.4	65.8	64.7($\downarrow$0.9)	63.8($\downarrow$1.6)	65.6($\downarrow$0.2)	64.7($\downarrow$0.9)	64.7($\downarrow$0.7)	64.7($\downarrow$1.1)
NWPU-Crowd	77.8	81.3	74.5	77.4($\downarrow$0.4)	80.9($\downarrow$0.4)	74.1($\downarrow$0.4)	77.0($\downarrow$0.8)	80.5($\downarrow$0.8)	73.6($\downarrow$0.9)

TABLE 8: Anchor redundancy on five datasets.

Dataset	STA	STB	UCF-QNRF	JHU-Crowd++	NWPU-Crowd
Anchor Redundancy (AR)	34%	21%	25%	23%	13%
F1 gain by AM	1.2	0.7	0.75	0.9	0.4

TABLE 9: Ablation study of different re-matching strategies. The best results are highlighted in red.

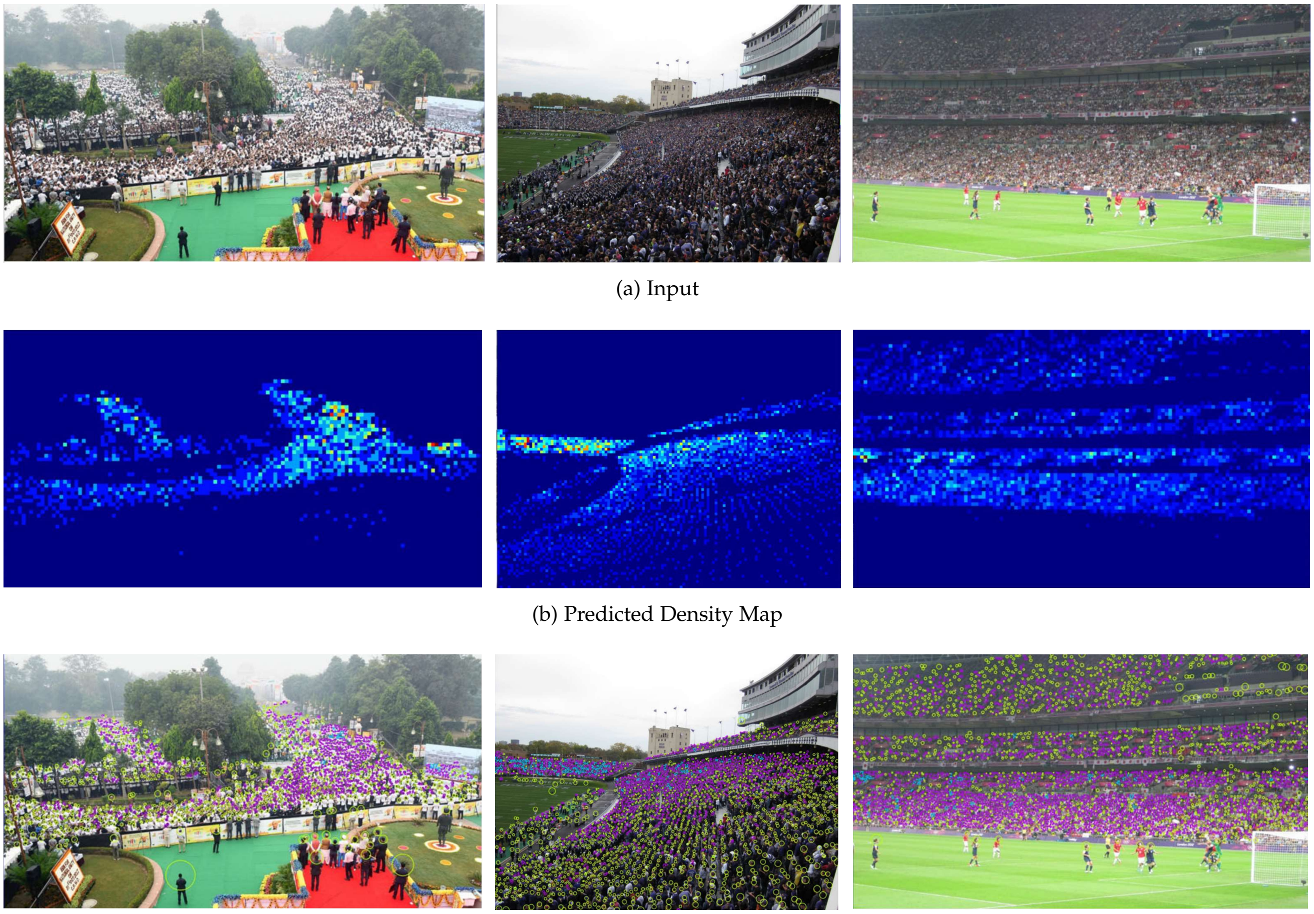
Stragety	$\sigma = 4$			$\sigma = 8$		
	F1↑	P↑	R↑	F1↑	P↑	R↑
None	37.8	37.3	38.3	64.7	63.8	65.6
DR	40.2	39.9	40.6	64.9	64.9	64.8
HCR	40.3	40.1	40.4	65.0	65.2	64.8
IP	40.4	40.1	40.8	65.6	65.4	65.8

the second-best strategy, HCR. This can be attributed to AM adopting the same metric used during inference.

Performance under different crowd density levels of JHU-Crowd++ dataset is presented in Table 10. It shows that the proposed AM strategy is more effective when the number of people is larger: the F1 score is improved by

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013



(a) Input

(b) Predicted Density Map

(c) Predicted Result

Fig. 9: Results of our CAAPN on images of the JHU-CROWD++ test dataset (image ids: 1795, 2592, and 3178). In (b), the density map's color intensifies with the increase in crowd density. In (c), circles in blue, yellow, and red denote crowd, medium, and sparse levels, respectively. The circle sizes indicate the estimated scale.

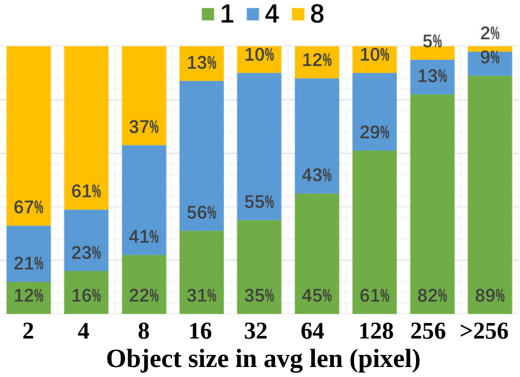


Fig. 10: The performance of different branches on different object sizes in terms of Recall (normalized to 1), tested on JHU-CROWD++.

TABLE 10: Ablation study of different crowd levels.

Count	with AM					w/o AM				
	F1↑	P↑	R↑	MAE↓	MSE↓	F1↑	P↑	R↑	MAE↓	MSE↓
[0,10]	95.7	93.5	97.9	1.2	3.1	95.6	93.3	97.9	1.1	3.2
(10,100]	89.1	88.2	90.0	10.9	45.6	88.7	87.8	89.6	11.2	46.5
(100-1000]	65.4	64.9	65.8	64.7	245.8	64.7	64.0	65.4	65.1	250.9
(1000,+∞)	54.2	53.1	55.3	105.1	443.7	52.5	51.6	53.4	107.2	455.7

TABLE 11: The crowd counting and localization performance comparison between [36] and our AM on the JHU-Crowd++ dataset. 1x, 2x, 3x means using 1x, 2x, 3x more queries, respectively. The best results are highlighted in red.

Method	MEM (G)↓	P ↑	R↑	F1↓	MAE↓	MSE ↓
Ours	22	65.4	65.8	65.6	60.8	250.9
[36] 1x	22	63.8	65.6	64.7	63.1	255.0
[36] 2x	31	64.2	65.8	65.0	62.5	250.9
[36] 3x	41	64.3	66.0	65.2	62.3	245.8

0.4, 0.7, and 1.7 for (10,100], (100,1000], and (1000,+∞), respectively. When the number of people is less than 10, the inconsistency issue is not serious, and the F1 improvement of AM degrades to 0.1. In summary, the AM strategy is effective in both dense and sparse scenarios, and it is more effective when dealing with images of large crowd density.

We further conduct experiments to test whether the many-to-one label assignment strategy used in object detection [36] can be transferred to crowd counting. The original method [36] uses 6 auxiliary heads and adds 6x queries to Deform DETR, which leads to too much memory cost when there are a large number of people in the crowd localization task. The number of objects in an image of its detection task is usually less than 100 but it is more than 500 for

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

TABLE 12: Ablation study of anchor pyramid levels. The best results are highlighted in red.

K	s_i	$\sigma = 4$	$\sigma = 8$	Counting
		F1/P/R	F1/P/R	MAE/MSE
1	1	37.9/38.1/37.7	62.9/63.3/62.5	61.9/262.0
	2	38.5/38.7/38.2	63.4/63.8/63.0	62.5/277.1
	4	38.7/39.3/38.1	64.7/64.7/64.7	63.7/272.1
	8	38.5/37.2/39.8	64.7/64.5/65.0	61.7/257.2
	10	38.4/37.2/39.7	64.6/63.9/65.2	61.0/254.1
2	1,4	39.7/40.1/39.5	65.0/65.2/64.8	61.0/264.0
	2,4	38.9/38.6/40.2	64.7/64.1/65.4	61.2/261.0
	4,8	39.2/38.1/40.5	64.0/63.3/64.8	64.2/272.0
3	1, 4, 8	40.4/40.1/40.8	65.6/65.4/65.8	60.8/250.9
	1, 2, 4	39.6/39.2/40.0	64.9/64.7/65.1	62.4/263.1
4	1,4,8,10	40.6/40.2/41.0	65.7/65.6/65.9	62.2/253.1
	1,4,8,12	40.7/40.2/40.9	65.8/65.6/65.9	63.1/255.2
	1,4,8,14	40.6/40.2/41.0	65.8/65.6/66.0	64.2/271.2
	1,4,8,16	40.7/40.2/41.1	65.9/65.7/66.1	64.5/267.3
5	1,4,8,16,32	40.5/40.1/40.9	65.5/65.2/66.8	65.0/271.2
6	1,4,8,16,32,64	40.3/39.5/41.0	65.2/63.9/66.5	65.9/280.0
7	1,4,8,16,32,64,128	39.9/39.0/40.8	64.6/63.1/66.1	67.2/294.7
10	1,4,8,16,32,64,128,256,512,1024	35.1/34.2/36.0	61.2/60.1/62.4	82.5/354.2

our crowd localization task. Thus, we change the number of queries to 1x, 2x, and 3x, respectively. The results on the JHU-Crowd++ dataset are shown in Table 11. It shows that our method performs best in almost all metrics, even the compared method using double memory as ours.

Effectiveness of APG. In Table 7, we present the performance of our method with and without APG. When removing the APG module, we set the number of anchors to 4 in a 16 × 16 grid, which is proved to be the best setting when using a fixed number of anchors, as shown in Table 12. The results in Table 7 show that our APG brings similar improvements on all datasets. In Figure 8, we present several qualitative comparisons between without using the APG module (a) and using APG (b). It shows that without the APG the model may predict too many points at the sparse regions (marked in the blue circle) and miss people in the crowded regions (marked in the red circle). In Figure 8(c), we present the anchors generated by our APG module.

Function of different locating branches. Our model contains three locating branches, which are equipped with 1, 4, and 8 anchors, respectively. Intuitively, the branch with fewer anchors is supposed to predict big-size targets. To verify this conjecture, we compute the percentage of predictions obtained by each branching under different target sizes. Similar to other experiments, we use the average edge length of a bounding box to denote its size. The results are shown in Figure 10(a). Targets with extremely large scales (the average edge length larger than 256 pixels), more than 89% are predicted by the branch with only one anchor. For tiny objects (the average edge length smaller than 2 pixels), 67% are generated by the branch with eight anchors.

Different Anchor Pyramid Level. In our methods, we learn anchor-priors under K density levels. Here we explore different K values, and for the same K we also investigate different density combinations. The results are presented in Table 12. It shows that when using only one pyramid level (*i.e.*, $K = 1$), setting the anchor number to 4 yields the best localization performance. When increasing K to 2, the best F1 with $\sigma = 4$ is further improved (38.7 v.s. 39.7). When $K = 4$, the model achieves the best localization

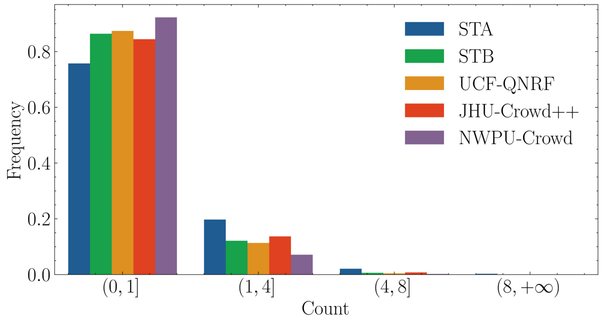


Fig. 11: The frequency of count in a 16 × 16 patch. The x-axis is the count, and the y-axis is the frequency.

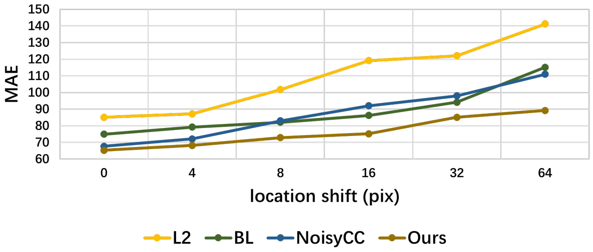


Fig. 12: Robustness of loss to location shift errors.

performance. However, we notice a significant drop in the counting performance. As shown in Figure 11, the count distribution in the training dataset within each 16 × 16 patch is unbalanced. Using K=3, patches with more than 4 people will be assigned to branches s_3. If we increase K, some patches will be assigned to other branches, exacerbating the imbalance problem and making s_3 not fully trained. In contrast, $K = 3$ with $s_1 = 1, s_2 = 4, s_3 = 8$ is a well trade-off between counting and localization performance.

TABLE 13: Ablation study of anchor-prior generation strategies.

Method	$\sigma = 4$	$\sigma = 8$	Counting
	F1/P/R	F1/P/ R	MAE/MSE
AAG	40.4/40.1/40.8	65.6/65.4/65.8	60.8/250.9
AAG w/o K-means	40.4/40.1/40.7	65.4/65.2/65.3	61.0/259.1
w/o AAG	38.7/39.3/38.1	64.7/64.7/64.7	63.2/271.0

Effectiveness of Anchor Spatial Distribution Prior We utilize the spatial distribution of targets for our anchors' positions in a region. To verify its effectiveness, we replace them with evenly distributed ones as in [9]. The results are presented in Table 13. With K-means, the prior brings 0.2 F1 improvements when setting $\sigma = 8$ (65.6 v.s. 65.4).

Effectiveness of CCL. To evaluate the efficacy of CCL, we replace our loss $\mathcal{L}_{ccl}$ with the conventional L2 loss. The results, as presented in Table 14, show that the use of L2 loss results in an 8.4 decrease in MAE (69.2 v.s. 60.8) and, more significantly, a 4.2 decrease in F1 ($\sigma = 8$). For CRL, it outperforms L1, BL, and NoisyCC, which are designed to address location shift error (or label noise) on both $\sigma = 4$ and $\sigma = 8$ for all metrics (F1, P, R, MAE, MSE). It outperforms L1, BL and NoisyCC on both $\sigma = 4$ and $\sigma = 8$ for all metrics (F1, P, R, MAE, MSE). With re-weighting, the F1 score is improved

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

TABLE 14: Ablation study of different counting losses used for training counting head.

Counting Loss	$\sigma = 4$	$\sigma = 8$	Counting
	F1/P/R	F1/P/ R	MAE/MSE
L1	38.7/38.2/39.3	61.2/60.5/62.0	69.2/293.9
L1+IL	38.9/38.3/39.5	61.5/60.9/62.1	70.5/294.5
BL	39.1/38.2/40.2	62.1/60.8/63.2	65.5/278.2
BL+IL	39.0/38.3/39.8	62.0/61.0/63.1	66.9/292.1
NoisyCC	39.6/39.4/49.8	64.5/64.1/64.9	63.7/271.0
NoisyCC+IL	39.8/39.5/50.1	65.0/64.7/65.3	64.9/288.4
CRL w/o re-weight	39.2/39.1/39.3	63.6/63.3/63.9	63.2/276.7
CRL	40.3/40.0/40.7	65.3/65.2/65.4	61.3/262.9
CRL+IL	40.4/40.1/40.8	65.6/65.4/65.8	60.8/250.9

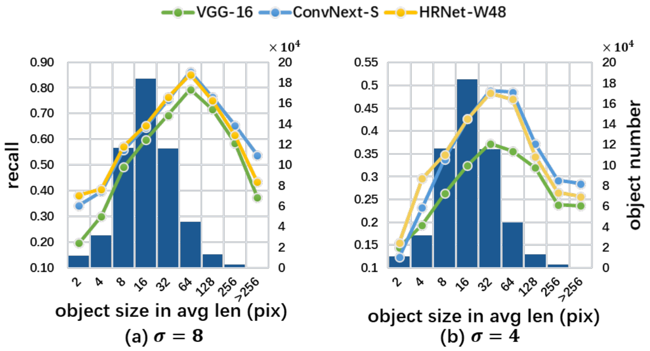
by 1.1 and 1.0 on $\sigma = 4$ and $\sigma = 8$, respectively. And even without re-weighting, CRL still outperforms L1 loss on both $\sigma = 4$ and $\sigma = 8$ as it is supervised on multi-resolution density maps. We also conduct more detailed ablation experiments on the components of CCL, specifically by removing L_{IL}. The inclusion of IL enhances the accuracy of quantity predictions (60.8 v.s. 61.3) as it compels the predictions to approximate integers, thereby reducing rounding errors. IL is designed to reduce the quantization error of the density map. It works well with density map based loss, including L1, NoisyCC, and our CRL, improving the F1 score by 0.3, 0.5, and 0.1, respectively. IL does not work well for the dot map-based loss BL, as the dot map is not sensitive to the quantization error [53]. To verify CCL's effectiveness in localization shift error further, we introduce noise to the point annotations of the JHU-Crowd++ training set. New annotations are evenly sampled within a circle, centered at each original annotation and with a radius of r, where r is 2, 4, 8, 16, 32, and 64respectively. The counting performance of these losses is shown in Figure 12. The performance of all losses deteriorates with the introduction of noise, but our method exhibits a slower rate of increase, especially when the shift error exceeds 16 pixels. This indicates the superiority of our loss function.

TABLE 15: The latency, parameters, and performance of different methods.

Method	Backbone	Parameters↓	Latency↓	F1↑
STEERER [58]	HRNet-W48	64.5M	63ms	77.0
GMS [49]	HRNet-W48	66.1M	55ms	78.1
FIDTM [18]	HRNet-W48	65.1M	53ms	75.5
CSRNet [50]	VGG16	14.6M	30ms	52.1
Ours	HRNet-W48	67.2M	58ms	78.6
Ours	VGG16	14.7M	34ms	76.5

Computation cost. We evaluate the counting and localization performance of our method with several existing methods [18], [49], [50], [58], the result is shown in Table 15. It shows that using the same backbone, our model has slightly more parameters and comparable latency. Considering the large performance improvements over these methods, the increased computation cost is acceptable.

Effect of different features. Figure 13 shows the localization performance curve with respect to different target sizes on the JHU-CROWD++ dataset (other datasets do not provide bounding box information) dataset, and presents the histogram of target size distribution. Similar to other experiments, the size of a target is measured by the average length of bounding box edges ((width+height)*0.5). The recall rate first is increased as the target size becomes larger, but the performance starts to drop when the average length is larger than 64 pixels for all three kinds of features. This is likely caused by the receptive fields being insufficient for such large objects.



(a) $\sigma = 8$

(b) $\sigma = 4$

Fig. 13: Localization performance of different features and object sizes (avg len $= \frac{h+w}{2}$). The histograms show the object quantity distribution under different sizes. Results are obtained on the JHU-CROWD++ dataset.

TABLE 16: Detection performance on the CrowdHuman dataset.

Method	Epoch	AP50↑	mMR↓	Recall↑
FCOS [33]	36	91	46.5	97.9
FCOS [33]+DDQ [37]	36	92.7	41.0	98.2
FCOS [33]+AM	36	93.0	39.5	98.4
Deform DETR	50	89.1	50.0	95.3
Deform DETR [32]+Hybird [36]	36	92.5	44.2	97.0
Deform DETR [32]+DDQ [37]	36	93.8	39.7	98.7
Deform DETR [32]+AM	36	89.5	46.2	96.1

Applying AM to object detection. The inconsistency problem also exists in query/anchor-based object detection methods [4], [30]–[32], which often use IoU and classification score together during training but only use classification score during inference. We apply our method to object detection and compare our AM strategy with [37] and [36] on their common evaluation detection dataset CrowdHuman, using two representative query-based object detection frameworks as the baseline: FCOS [33] (dense queries) and Deform DETR [32] (sparse queries). The results are shown in Table 16. It shows that our AM strategy can improve FCOS and Deform DETR performance with the same or fewer epochs. On the dense queries baseline FCOS, the improvement is 2.0, 7.0, and 0.5 in terms of AP50, mMR, and Recall, respectively, showing advantages compared to [37]. We also note that on the sparse queries baseline Deform DETR, the improvement is 0.4, 3.8, and 0.8 on AP50, mMR, and Recall, respectively. Our improvements are less than [37] and [36]. This is because, in sparse query-based methods, the queries are less redundant than in dense query-based methods, making the inconsistency problem less severe. Therefore, the improvement of our AM strategy on sparse query-based object detection methods is not very significant.

5 Conclusions

We propose a novel consistency-aware anchor pyramid network for crowd localization by predicting the precise

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

locations of human heads. The model consists of two key components: an Adaptive Anchor Generator (AAG) and a Localizer with Augmented Matching (LAM). The AAG adaptively determines the anchor density in each image region based on the predicted crowd density. This AAG takes the prior spatial distribution of heads into consideration, thus rendering the generated anchors more representative. The LAM then reduces the ranking inconsistency of predictions during training and inference. Evaluated with three kinds of popular features, i.e. VGG-16, HRNet-W48, and ConvNeXt-S, our method achieves superior performance against existing methods on five widely-used benchmarks with diverse crowd densities and scenes.

References

[1] Z. Zhao, H. Li, R. Zhao, and X. Wang, "Crossing-line crowd counting with two-phase deep neural networks," in *ECCV*, 2016, pp. 712–726. 1

[2] Z. Huang, Y. Ding, G. Song, L. Wang, R. Geng, H. He, S. Du, X. Liu, Y. Tian, Y. Liang *et al.*, "Bcdata: A large-scale dataset and benchmark for cell detection and counting," in *MICCAI*, 2020, pp. 289–298. 1

[3] A. Aldayri and W. Albattah, "Taxonomy of anomaly detection techniques in crowd scenes," *Sensors*, vol. 22, no. 16, p. 6080, 2022. 1

[4] J. Wang, L. Song, Z. Li, H. Sun, J. Sun, and N. Zheng, "End-to-end object detection with fully convolutional network," in *CVPR*, 2021, pp. 15 849–15 858. 1, 3, 13

[5] X. Chu, A. Zheng, X. Zhang, and J. Sun, "Detection in crowded scenes: One proposal, multiple predictions," in *CVPR*, 2020, pp. 12 214–12 223. 1, 2

[6] J. Wan, Z. Liu, and A. B. Chan, "A generalized loss function for crowd counting and localization," in *CVPR*, 2021, pp. 1974–1983. 1

[7] D. Lian, J. Li, J. Zheng, W. Luo, and S. Gao, "Density map regression guided detection network for rgb-d crowd counting and localization," in *CVPR*, 2019, pp. 1821–1830. 1, 2

[8] D. Lian, X. Chen, J. Li, W. Luo, and S. Gao, "Locating and counting heads in crowds with a depth prior," *IEEE TPAMI*, 2021. 1, 2

[9] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Wu, "Rethinking counting and localization in crowds: A purely point-based framework," in *ICCV*, 2021, pp. 3365–3374. 1, 2, 3, 4, 5, 6, 7, 8, 12

[10] D. Liang, W. Xu, and X. Bai, "An end-to-end transformer model for crowd localization," in *ECCV*, 2022, pp. 38–54. 1, 2, 3, 5, 6, 8

[11] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Bayesian loss for crowd count estimation with point supervision," in *ICCV*, 2019, pp. 6142–6151. 1, 2, 8

[12] B. Wang, H. Liu, D. Samaras, and M. H. Nguyen, "Distribution matching for crowd counting," in *NeurIPS*, 2020, pp. 1595–1607. 1, 2, 8

[13] S. Abousamra, M. Hoai, D. Samaras, and C. Chen, "Localization in the crowd with topological constraints," in *AAAI*, 2021. 1, 2, 6, 7, 8

[14] J. Gao, T. Han, Y. Yuan, and Q. Wang, "Learning independent instance maps for crowd localization," *arXiv preprint arXiv:2012.04164*, 2020. 1, 2, 6, 7

[15] C. Arteta, V. Lempitsky, and A. Zisserman, "Counting in the wild," in *ECCV*, 2016, pp. 483–498. 1, 2

[16] T. Han, J. Gao, Y. Yuan, X. Li *et al.*, "Ldc-net: A unified framework for localization, detection and counting in dense crowds," *arXiv preprint arXiv:2110.04727*, 2021. 1

[17] K. Kim and H. S. Lee, "Probabilistic anchor assignment with iou prediction for object detection," in *ECCV*, 2020, pp. 355–371. 1

[18] D. Liang, W. Xu, Y. Zhu, and Y. Zhou, "Focal inverse distance transform maps for crowd localization," *IEEE TMM*, 2022. 1, 3, 6, 7, 8, 13

[19] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *CoRR*, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767 2, 5

[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NeurIPS*, 2015. 2

[21] P. Hu and D. Ramanan, "Finding tiny faces," in *CVPR*, 2017. 2, 6

[22] Y. Liu, M. Shi, Q. Zhao, and X. Wang, "Point in, box out: Beyond counting persons in crowds," in *CVPR*, 2019, pp. 6469–6478. 2

[23] D. B. Sam, S. V. Peri, M. N. Sundararaman, A. Kamath, and R. V. Babu, "Locate, size, and count: Accurately resolving people in dense crowds via detection," *IEEE TPAMI*, vol. 43, no. 8, pp. 2739–2751, 2021. 2, 6, 8

[24] Y. Wang, J. Hou, X. Hou, and L.-P. Chau, "A self-training approach for point-supervised object detection and counting in crowds," *IEEE TIP*, vol. 30, pp. 2876–2887, 2021. 2

[25] J. Ribera, D. Güera, Y. Chen, and E. J. Delp, "Locating objects without bounding boxes," in *CVPR*, 2019, pp. 6472–6482. 2, 6

[26] 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 *ECCV*, 2018, pp. 532–546. 2, 6

[27] J. Cheng, H. Xiong, Z. Cao, and H. Lu, "Decoupled two-stage crowd counting and beyond," *IEEE TIP*, vol. 30, pp. 2862–2875, 2021. 2

[28] W. Lin and A. B. Chan, "Optimal transport minimization: Crowd localization on density maps for semi-supervised counting," in *CVPR*, 2023, pp. 21 663–21 673. 2

[29] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez, and M. Schmidt, "Where are the blobs: Counting by localization with point supervision," in *ECCV*, 2018, pp. 547–562. 2

[30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *ECCV*, 2020, pp. 213–229. 3, 8, 13

[31] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang *et al.*, "Sparse r-cnn: End-to-end object detection with learnable proposals," in *CVPR*, 2021, pp. 14 454–14 463. 3, 13

[32] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai, "Deformable detr: Deformable transformers for end-to-end object detection," in *ICLR*, 2020. 3, 13

[33] Z. Tian, C. Shen, H. Chen, and T. He, "Fcos: Fully convolutional one-stage object detection," in *ICCV*, 2019, pp. 9627–9636. 3, 13

[34] S. Zhang, C. Chi, Y. Yao, Z. Lei, and S. Z. Li, "Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection," in *CVPR*, 2020. 3, 5

[35] K. Kim and H. S. Lee, "Probabilistic anchor assignment with iou prediction for object detection," in *ECCV*, 2020, pp. 355–371. 3

[36] Z. Zong, G. Song, and Y. Liu, "Detrs with collaborative hybrid assignments training," in *ICCV*, 2023, pp. 6748–6758. 3, 11, 13

[37] S. Zhang, X. Wang, J. Wang, J. Pang, C. Lyu, W. Zhang, P. Luo, and K. Chen, "Dense distinct query for end-to-end object detection," in *CVPR*, 2023, pp. 7329–7338. 3, 13

[38] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Journal of the royal statistical society. series c (applied statistics)*, vol. 28, no. 1, pp. 100–108, 1979. 4

[39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *CVPR*, 2015, pp. 1–9. 4, 8

[40] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang *et al.*, "Deep high-resolution representation learning for visual recognition," *IEEE TPAMI*, vol. 43, no. 10, pp. 3349–3364, 2020. 4

[41] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *CVPR*, 2022, pp. 11 976–11 986. 4

[42] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016, pp. 21–37. 5

[43] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *CVPR*, 2017, pp. 2117–2125. 5

[44] H. W. Kuhn, "The hungarian method for the assignment problem," *NRL*, vol. 2, no. 1-2, pp. 83–97, 1955. 5

[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *ICCV*, 2017, pp. 2980–2988. 5, 7

[46] C. Liu, X. Weng, and Y. Mu, "Recurrent attentive zooming for joint crowd counting and precise localization," in *CVPR*, 2019, pp. 1217–1226. 6, 7, 8

[47] C. Xu, D. Liang, Y. Xu, S. Bai, W. Zhan, X. Bai, and M. Tomizuka, "Autoscale: Learning to Scale for Crowd Counting," *IJCV*, vol. 130, no. 2, pp. 405–434, 2022. 6, 8

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2024.3392013

[48] J. Gao, M. Gong, and X. Li, "Congested crowd instance localization with dilated convolutional swin transformer," *Neurocomputing*, vol. 513, p. 94–103, 2022. 6, 7

[49] J. Wang, J. Gao, Y. Yuan, and Q. Wang, "Crowd localization from gaussian mixture scoped knowledge and scoped teacher," *IEEE TIP*, vol. 32, pp. 1802–1814, 2023. 6, 8, 13

[50] Y. Li, X. Zhang, and D. Chen, "Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes," in *CVPR*, 2018, pp. 1091–1100. 8, 13

[51] V. Sindagi and V. M. Patel, "Multi-Level Bottom-Top and Top-Bottom Feature Fusion for Crowd Counting," in *ICCV*, 2019, pp. 1002–1012. 8

[52] Y. Hu, X. Jiang, X. Liu, B. Zhang, J. Han, X. Cao, and D. Doermann, "Nas-count: Counting-by-density with neural architecture search," in *ECCV*, 2020, pp. 747–766. 8

[53] J. Wan and A. Chan, "Modeling noisy annotations for crowd counting," in *NeurIPS*, 2020. 8, 13

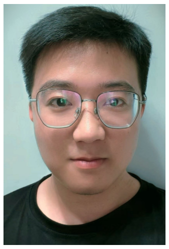
[54] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *CVPR*, 2016, pp. 589–597. 6

[55] V. A. Sindagi, R. Yasarla, and V. M. Patel, "Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method," *IEEE TPAMI*, vol. 44, no. 05, pp. 2594–2609, 2022. 6

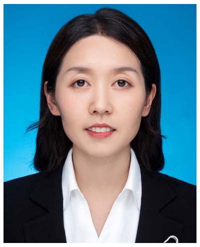
[56] Q. Wang, J. Gao, W. Lin, and X. Li, "Nwpu-crowd: A large-scale benchmark for crowd counting and localization," *IEEE TPAMI*, vol. 43, no. 6, pp. 2141–2149, 2020. 6

[57] R. Wightman, "Pytorch image models," https://github.com/rwightman/pytorch-image-models, 2019. 7

[58] T. Han, L. Bai, L. Liu, and W. Ouyang, "Steerer: Resolving scale variations for counting and localization via selective inheritance learning," in *ICCV*, 2023, pp. 21 848–21 859. 13



Xinyan Liu received the B.S degree from Harbin Institute of Technology (Weihai) in 2019 and now pursuing his Ph.D. degree in University of Chinese Academy of Science. His research interests include crowd counting and localization, class agnostic counting, and video language tracking.



Guorong Li (Senior Member, IEEE) received the B.S. degree in technology of computer application from the Renmin University of China, China, in 2006, and the Ph.D. degree in technology of computer application from the Graduate University of Chinese Academy of Sciences in 2012. She is currently an Associate Professor at the University of Chinese Academy of Sciences. Her research interests include video analysis, pattern recognition, and cross-media analysis.

Yuankai Qi received the M.S. and Ph.D. degrees from Harbin Institute of Technology, China, in 2013 and 2018, respectively. His research interests include vision-language navigation, object tracking, video captioning, and visual voice cloning. He served as area chair for IJCAI 2021 and BMVC 2023. He is also a regular reviewer for top-tier venues, such as IEEE TPAMI, TIP, CVPR, NeurIPS, ICCV, AAAI, etc.

Zhenjun Han received the B.S. degree in software engineering from Tianjin University, Tianjin, China, in 2006 and the M.S. and Ph.D. degrees from University of Chinese Academy of Sciences, Beijing, China, in 2009 and 2012, respectively. Since 2013, he has been an Associate Professor with the School of Electronic, Electrical, and Communication Engineering, the University of Chinese Academy of Sciences. His research interests include object tracking and detection.

Anton van den Hengel is the founding Director of The Australian Institute for Machine Learning, a Chief Investigator of the Australian Centre of Excellence in Robotic Vision, and a Professor of Computer Science at the University of Adelaide.

Nicu Sebe (SM'11) is Professor in the University of Trento, Italy, where he is leading the research in the areas of multimedia analysis and human behavior understanding. He was the General Co-Chair of the IEEE FG 2008 and ACM Multimedia 2013. He was a program chair of ACM Multimedia 2011 and 2007, ECCV 2016, ICCV 2017 and ICPR 2020 and a general chair of ACM Multimedia 2022. He is a fellow of IAPR.

Ming-Hsuan Yang (Fellow, IEEE) is a professor in electrical engineering and computer science at the University of California, Merced, Merced, California. He serves as a program co-chair of IEEE International Conference on Computer Vision (ICCV), in 2019, program co-chair of the Asian Conference on Computer Vision (ACCV), in 2014, and general co-chair of ACCV 2016. Yang served as an associate editor of the IEEE Transactions on Pattern Analysis and Machine Intelligence from 2007 to 2011 and is an associate editor of the International Journal of Computer Vision, the Image and Vision Computing, and the Journal of Artificial Intelligence Research. He received the NSF CAREER award, in 2012 and Google Faculty Award, in 2009. Yang is a Fellow of the IEEE and ACM.

Qingming Huang (Fellow, IEEE) received the bachelor's degree in computer science and the Ph.D. degree in computer engineering from the Harbin Institute of Technology, China, in 1988 and 1994, respectively. He is currently a Professor with the University of Chinese Academy of Sciences and an Adjunct Research Professor with the Institute of Computing Technology, Chinese Academy of Sciences. He has authored or coauthored more than 400 papers in prestigious international journals and conferences. His research interests include multimedia computing, image processing, computer vision, and pattern recognition. He is an Associate Editor of IEEE TCSVT and *Acta Automatica Sinica*. He has served as the General Chair, the Program Chair, the Track Chair, and a TPC Member for various conferences, including ACM MM, CVPR, ICCV, ICME, ICM.

Authorized licensed use limited to: Univ of Calif Merced. Downloaded on February 03,2025 at 04:06:13 UTC from IEEE Xplore. Restrictions apply.
© 2024 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.