The Visual Computer (2024) 40:1053–1067
https://doi.org/10.1007/s00371-023-02831-z

ORIGINAL ARTICLE

CLFormer: a unified transformer-based framework for weakly supervised crowd counting and localization

Mingfang Deng[1] · Huailin Zhao[1] · Ming Gao[1]

Accepted: 1 March 2023 / Published online: 12 April 2023

Abstract
Recent progress in crowd counting and localization methods mainly relies on expensive point-level annotations and convolutional neural networks with limited receptive filed, which hinders their applications in complex real-world scenes. To this end, we present CLFormer, a Transformer-based weakly supervised crowd counting and localization framework. The model extracts global information from the input image using a Transformer and then passes the extracted features to both a regression branch for crowd counting and a localization branch for localization. Initial proposals are produced by the localization branch and filtered via score maps generated from the extracted features, and their centers are used as pseudo-point-level annotations. Through staggered training of the two branches, the quality of pseudo-point-level annotations is improved, and the final localization maps are generated. Experiments on four benchmark datasets (i.e., ShanghaiTech, UCF-QNRF, JHU-CROWD++, and NWPU-Crowd) demonstrate that CLFormer obtains better counting performance than weakly supervised and fully supervised counting networks and comparable localization performance to fully supervised localization networks.

Keywords Shunted Transformer · Weakly supervised learning · Crowd counting · Crowd localization

1 Introduction

Crowd counting is a classical computer vision task that is to estimate the number of people in an image or video frame. It is particularly prominent because of its special significance for public safety, urban planning and metropolitan crowd management [1].

In recent years, convolutional neural network-based methods [2–7] have achieved significant progress in crowd counting task. They formulate the task as a regression problem and design sophisticated networks to learn the nonlinear relationship between the input crowd image and its corresponding crowd density map. For example, MCNN [2] adopted a multi-column convolutional neural network with different convolutional structures to capture the scale variations of crowd heads. Zhang et al. [7] proposed a novel method named two-task convolutional neural network to simultaneously learn two tasks that are dense degree classification and density map estimation, respectively. However, these regression methods require point-level annotations to generate ground truth density maps, which are often labor-intensive and time-consuming.

To solve this limitation, some works leverage weakly supervised learning paradigms to perform crowd counting. These researchers used deep network models to establish a nonlinear mapping between input images and their corresponding counts of images. For example, MATT [8] learns models from a few point-level annotations (fully supervised) and a large number of count-level annotations (weakly supervised). Yang et al. [9] designed a soft label sorting network to regress images directly to predict the crowd counts. However, due to the locality property in the convolution operation, the above methods cannot directly obtain the global information of the input crowd scene, which is beneficial to crowd counting in the complex scene.

In recent years, inspired by the success of Transformer [10], some researchers intended to introduce the Transformer models into the weakly supervised crowd counting field. For example, TransCrowd [11] adopted the vision

✉ Huailin Zhao
tyoukr@163.com

Mingfang Deng
dengmingfang2021@163.com

Ming Gao
gaoming_one@163.com

[1] School of Electrical and Electronic Engineering, Shanghai Institute of Technology, Shanghai 201418, China

Transformer [12] (ViT) model directly as a backbone network and regressed the predicted counts directly, resulting in a significant improvement in weakly supervised counting performance. However, there are some limitations to the application of the TransCrowd [11] directly for crowd counting. (1) ViT [12] models usually partition the image into non-overlapped and large size patches. But such partitioning is too coarse to lose much fine-grained information which will influence the precision of crowd counting. (2) The computational complexity of self-attention in the ViT [12] model is a square multiple of the number of tokens, which results in a large computational load. (3) ViT [12] models ignore multi-scale information in the scene.

In order to solve the problem of being unable to capture multi-scale information, some researchers design some Transformer architecture that can capture multi-scale information. For example, Pyramid Vision Transformer [13] (PVT) designs a spatially sparse self-attention for merging keys and query tokens, but PVT [13] tended to merge too many tokens in this space reduction, which made the detailed information of small objects mixed with the background and impaired the performance of the model. Ren et al. [14] proposed shunted self-attention, which can retain both coarse-grained and fine-grained information and maintain global dependency on image tokens. Motivated by the Shunted Transformer, we design crowd counting network consisting of four Shunted Transformer Blocks [14] (STBs), followed by extracting and fusing information from the four STBs, and finally regressed the predicted counts by a global average pooling.

In addition to crowd counting, crowd localization [15–17] is also a challenging task. In recent years, mainstream crowd localization methods have usually applied convolutional neural networks (CNNs) to generate density maps, followed by non-maximum suppression to obtain predicted localization maps. However, these crowd localization methods still rely on point-level annotations, which also face the same time-consuming and labor-intensive problem. Thus, a few scholars have proposed a crowd localization method based on count-level annotations (weakly supervised). For example, LOOC [16] achieves accurate crowd localization based on count-level annotations only by training the CNN to generate pseudo-point-level annotations.

Aiming to address the limitations of convolutional neural networks (CNNs) and inspired by LOOC [16], we propose CLFormer, a Transformer-based approach to weakly supervised crowd counting and localization. Specifically, we extract feature maps by fusing the results of the first and last STBs. We generate a certain number of initial proposals, which are then filtered by score maps derived from the extracted feature maps. The filtered proposals are identified as the foreground and their centers are identified as pseudo-point-level annotations. We update these foregrounds until the number of pseudo-point-level annotations is equal to the predicted counts. We further continuously train our networks to improve the quality of the pseudo-point-level annotations, eventually generating the final localization maps. To evaluate the effectiveness of our approach, we conducted extensive experiments on four publicly available datasets: ShanghaiTech [18], UCF-QNRF [19], JHU-CROWD++ [20], and NWPU-Crowd [15]. Our results demonstrate that our CLFormer achieves promising counting and localization performance.

In a summary, our contributions are twofold:

- CLFormer implements weakly supervised crowd counting and crowd localization tasks using Transformer architecture firstly.
- For the counting task, CLFormer achieves higher accuracy than the fully supervised counting methods and the weakly supervised counting methods, while for the localization task, comparable accuracy to fully supervised localization is achieved.

2 Related work

2.1 Fully supervised crowd counting

Fully supervised crowd counting methods use point-level annotations as a supervised signal to train a sophisticated counting network to predict count number in the images.

With the rapid development of CNNs, a number of multi-scale learning models have been proposed to solve the problem of multiple scales of human heads. For example, Zhang et al. [2] proposed a multi-column convolutional neural network (MCNN) with different convolutional structures to model the scale variations of crowd heads. Ma et al. [21] used unbalanced optimal transport (UOT) distance to quantify the discrepancy between two measures, outputting sharper density maps. Li et al. [22] proposed MSFFA which fuses different density classes of density maps to capture multi-scale information. SD et al. [23] propose an end-to-end scale invariant head detection framework that can handle broad range of scales. Wang et al. [24] propose an efficient crowd counting neural architecture search (ECCNAS) framework to search efficient crowd counting network structures. Simultaneously, some researchers used perspective information to alleviate the effect of scale variations. For example, Shi et al. [25] proposed a novel generating ground truth perspective maps strategy and predicted both the perspective maps and density maps at the testing phase. Yang et al. [26] proposed a reverse perspective network to estimate the perspective factor of the input image and then warp the image to solve the problem of severe occlusion. All of the above works are implemented based on the CNN.

However, CNN have limitations on the size of the convolutional kernel, which can reduce the ability to extract the human heads and impact the accuracy of counting. In contrast to CNN, Transformer models provide a global receptive field and have achieved significant progresses on many computer vision tasks [27–33]. Inspired by the success of Transformer, some researchers intended to introduce the Transformer models into the crowd counting area. For example, CCTrans [34] used Twins [35] as its backbone network and captures multi-scale information by applying differential dilated convolutional layers. Li et al. [36] utilized the swin Transformer to alleviate the problem of uneven distribution of crowd density. Lin et al. [37] designed a learnable region self-attention, which replaced the self-attention block in the vision Transformer [12] model with two different self-attention blocks that are learnable region attention and local attention regularization which can generate a more accurate density map. Usman et al. [38] introduced the notion of auxiliary and explicit image patch-importance ranking (PIR) and patch-wise crowd estimate (PCE) information to produce a third (run-time) modality.

2.2 Weakly supervised crowd counting

Weakly supervised crowd counting uses count-level annotations or few point-level annotations as a supervised signal and trains sophisticated counting networks to obtain predicted counts in the images. For example, MATT [8] learned a model from a small amount of point-level annotations and a large amount of count-level annotations. Liu et al. [39] used the similarity between individuals in each image to directly supervise unlabeled regions. Wang et al. [11] directly regressed the global counts, and some negative samples are fed into the network to boost the robustness. Yang et al. [9] also directly mapped the images to the crowd numbers without point-level annotations based on the proposed soft-label sorting network. Similarly, in order to better extract contextual information, some scholars have introduced Transformer models into the weakly supervised counting fields. For example, TransCrowd [11] used the ViT model [12] and directly regressed the predicted counts by global average pooling. Compared with these popular weakly supervised crowd counting, our designed model adopts different scales of Transformer encoder to effectively capture the multi-scale information in the images.

2.3 Fully supervised crowd localization

Fully supervised crowd localization methods use point-level annotations as a supervised signal and train sophisticated localization networks to obtain the precise location of each object in the images. These methods can be broadly divided into three categories: detection-based localization methods [40,41], density map-based localization methods [42] and regression-based localization methods [42], respectively. In the early period, some researchers used detection-based localization methods that is mainly to design an algorithm to detect certain salient parts of the human body. For example, PSDDN [43] used a curriculum learning strategy for crowd localization, using the nearest neighbor head distance to initialize the position of each person candidate box, and the network was continuously trained to accurately determine the position of each person candidate box. LSC-CNN [41] also LSC-CNN adopted a multi-column architecture with a top-down approach to better solve the multi-scale problem, generated candidate box of each person and proposed a new winner-take-all loss for better training at higher resolutions. Liang et al. [42] introduce a KMO-based Hungarian matcher, which adopts the nearby context as the auxiliary matching cost to realize the crowd localization.

In recent years, a large number of researchers have also adopted density map-based localization methods that obtain the predicted localization of each person in the image by Hungarian algorithm and non-maximal suppression. For example, Idress et al. [44] and Gao et al. [17] designs an adaptive Gaussian kernel which defines the individual Gaussian kernel as the minimum of the Euclidean distance of each person, so that the position information of each person is more accurate by solving the multi-scale and severe occlusion problems. Even though using the adaptive Gaussian kernel can generate sharp density maps, it did not pay enough attention to the appearance of the person features, and when the scene changes, the performance will be seriously reduced. To solve this issue, some methods focus on designing new density maps to address the impact of complex backgrounds, such as the focal inverse distance transform map [45] (FIDTM), distance label map [46]. These methods can effectively avoid overlap in the dense regions, but they need post-processing to extract the instance location and rely on multi-scale feature maps, which are not highly efficient. Recently, in order to improve localization efficiency and accuracy, some studies have adopted regression-based methods. For example, P2PNet [47] is a regression-based framework for crowd localization relying on pre-processing that is by defining surrogate regression on a large set of proposals to predict the location of each person.

2.4 Weakly supervised crowd localization

The weakly supervised crowd localization methods train sophisticated localization networks and employ count-level annotations as supervised signals to obtain the exact localization of each object in the image. Most weakly supervised localization methods fall under multiple-instance learning (MIL) [48]. In this setup, each image corresponds to a certain number of object proposals, these object proposals are

labeled based on whether an object class exists. C-WSL [49] is the most relevant to our work as they use count information to obtain the highest-scoring proposals. However, it is not suitable for localization tasks in dense scenes. In order to solve the limitations of the above scenes, LOOC [16] only relied on count-level annotations to learn the occlusion object localization task and the model is suitable for dense scenes. However, the above methods are based on CNN models. In contrast, we try to capture more contextual information by using the Transformer architecture to achieve more accurate weakly supervised crowd localization.

3 Method

CLFormer is mainly comprised of the encoder branch (Transformer) which is used to extract features, the regression branch and the localization branch, as shown in Fig. 1. Specifically, given an input image, it is firstly divided into fixed-size patches and then flatten into a sequence of column vectors. The sequence is fed into the Transformer, followed by the regression branch to produce the predicted counts. Meanwhile, we fuse the final feature map and the middle feature map of the Transformer, followed by the localization branch to generate the localization map. The regression branch uses the ground truth corresponding to the input image as the supervised signal, while the localization branch uses the predicted counts followed by the regression branch as the supervised signal. We train the regression branch and the localization branch alternately until the number of pseudo-point annotations from the localization branch is equal to the number of predicted counts from the regression branch that are the final localization maps. Next, we will first introduce the details of Transformer we used in CLFormer and, then, the details of the counting task and the localization task in Sects. 3.3 and 3.4, respectively.

3.1 Problem formulation

Weakly supervised crowd counting definition. We formulate weakly supervised crowd counting following [8,11]. The input image I is fed into the counting network, and obtained feature maps are regressed to the predicted counts $\hat{C}_i$ by global average pooling. To be specific, the predicted counts $\hat{C}_i$ is formulated as follows:

$$\begin{aligned} \mathrm{FM} &= \mathcal{F}(I_i), \\ \hat{C}_i &= Pool2D(FM), \end{aligned} \tag{1}$$

where $\mathcal{F}(\cdot)$ stands for the counting network. $\hat{C}_i$ represents the predicted counts of the i-th input image I_i. $Pool2D(\cdot)$ represents the global average pooling. FM represents the feature map after the counting network.

The counting network learns the deviation between predicted counts and ground truth counts of the i-th image. We choose $\mathcal{L}_1$ loss to optimize our network. The loss function is defined as:

$$\mathcal{L}_1 = \frac{1}{N}\sum_{i=1}^{N} |C_i - \hat{C}_i|, \tag{2}$$

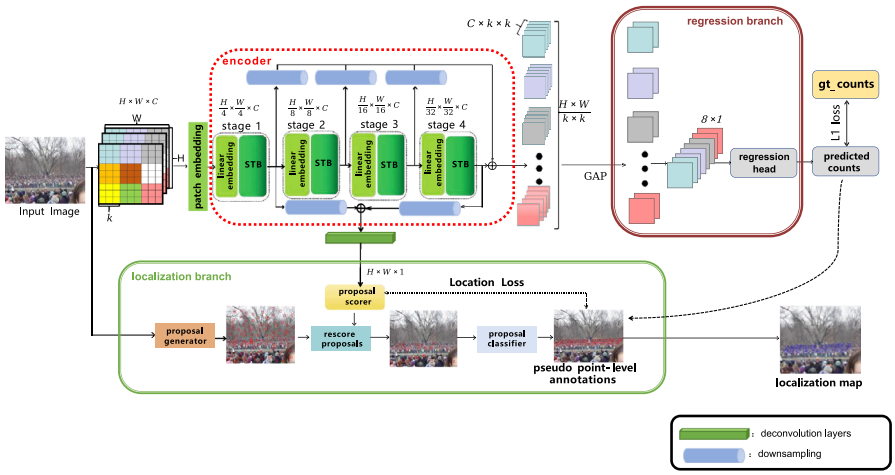

Fig. 1 The structure of CLFormer. CLFormer is composed of two branches: the regression branch and the localization branch. In order to reduce the computational effort, the regression branch first uses the average global pooling operation (GAP) to further extract the features and then uses the regression head to obtain the predicted counts. The localization branch adopts the proposal generator to generate the initial proposals and obtain the score of each proposal by the proposal scorer, and we send these rescore proposals to proposal classifier to produce the pseudo-point-level annotations. Finally, when the number of pseudo-point-level annotations equals the predicted count, we obtain the localization map

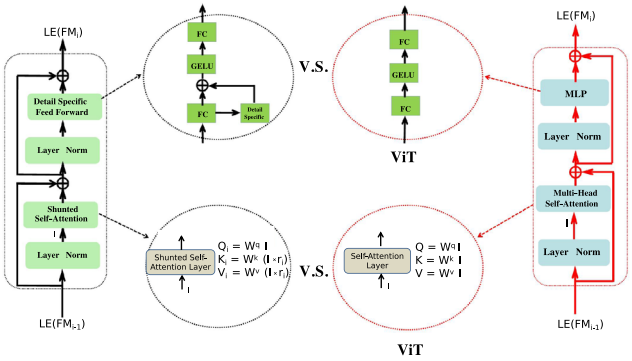

Fig. 2 The leftmost dashed box is the structure of i-th STB. $LE(\cdot)$ is linear embedding. The FM_i represents the output of the i-th stage. And we compare the feed-forward layer and self-attention layer in shunted Transformer (black circles) between ViT (red circles)

where C_i represents the ground truth counts of the i-th input image and the N stands for the number of images.

Weakly supervised crowd localization definition. Weakly supervised crowd localization methods [16,50] only use count-level annotations as a supervised signal and train the sophisticated localization networks $\mathcal{H}$ to obtain the location of each object in the images. To be specific, the process of obtaining the position of each object is formulated as follows:

$$\begin{aligned} P &= \text{sel}(I_i), \\ \text{CAM} &= \mathcal{H}(I_i), \\ \mathcal{Y} &= \text{CAM}(P), \end{aligned} \tag{3}$$

where the $\text{sel}(\cdot)$ represents the selective search method to produce the initial proposals. CAM represents the score map. The $\mathcal{Y}$ are part of the location of each object in the images.

3.2 Transformer

Image to sequence. The first step is to transform the input image $I \in \mathbb{R}^{H\times W\times 3}$ into a sequence of 2D flattened patches. Specifically, we reshape the image I into N patches, resulting in $X = \{x_n \in \mathbb{R}^{k\times k\times 3} \mid n = 1, 2, 3, ..., N\}$, where $N = \frac{H}{k} \times \frac{W}{k}$, and k is the patch size.

Patch embedding. We map X into a latent D-embedding feature with a learnable projection $x_n \in \mathbb{R}^D$. To obtain positional information for each patch, we need to add the position information P. The position information adopts learned positional embedding method. Thus, $P = \{P_n \in \mathbb{R}^N \mid n = 1, 2, 3, \ldots, N\}$, the patch embedding is formulated as:

$$\begin{aligned} E &= [e_1; e_2; e_3; \ldots; e_N] \\ &= [x_1 + P_1; x_2 + P_2; \ldots; x_N + P_N], \end{aligned} \tag{4}$$

Shunted Transformer block.

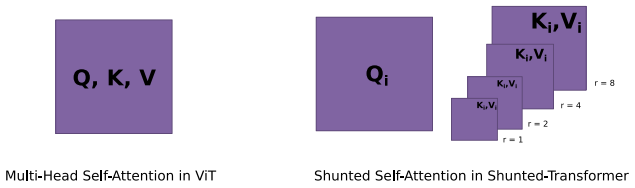

Fig. 3 Comparing shunted self-attention with self-attention in ViT and shunted self-attention in Shunted Transformer

In order to effectively capture the multi-scale information, we leverage the Transformer model containing the part of the Shunted Transformer [14] that has the different scales of K and V. The details of the Shunted Transformer block are shown in Fig. 2. Each Shunted Transformer block consists of shunted self-attention (SSA), and detail specific feedforward. The input sequence E is projected into query Q, key K and value V at first. Then, the multi-head self-attention (MSA) with H heads to compute self-attention operation in parallel. It is worth noting that the ViT [12] has two residual blocks, but the Shunted Transformer block does not have because Shunted Transformer block introduce a shunted attention mechanism for each self-attention layer to capture multi-granularity information and better model objects with different sizes than the two residual blocks in ViT [12]. Besides, different from the MSA in ViT [12], the key K and value V of SSA are down-sampled to different spatial sizes for different heads:

$$\begin{aligned} &Q_i = EW_i^Q, i \in \{1, 2, 3, 4\} \\ &K_i, V_i = \text{MTA}(E, r_i)W_i^K, \text{MTA}(E, r_i)W_i^V, \quad i \in \{1, 2, 3, 4\} \\ &V_i = V_i + \text{LE}(V_i), i \in \{1, 2, 3, 4\}, \end{aligned} \tag{5}$$

where $\text{MTA}(\cdot, r_i)$ is the multi-scale token aggregation layer in the head with the down-sampling rate of $r_i \in \{1, 2, 4, 8\}$. W_i^Q, W_i^K, W_i^V are the parameters of the linear projection in the i-th head. $\text{LE}(\cdot)$ is the local enhancing component of MTA for value V by a depth-wise convolution.

The shapes of Q, K and V are shown in the black dashed circle in the lower left corner of Fig. 3. The output h_i of SSA is as follows:

$$h_i = \text{Softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_h}}\right) V_i, \tag{6}$$

where d_h is the dimension. The K_i and the V_i are the i-th values of K and V.

Then, the feed forward layer of STB has the detailed specific (DS) layer. The fully connected layer is point-wise and no cross token information can be learnt in the traditional feed forward layer. Here, we utilize the DS to complement local information by specifying the details in the feedforward layer. As shown in the black dashed circle in the

upper left corner of Fig. 3. The formulas of the STB are as follows:

$$\begin{aligned} \mathcal{Z} &= \text{Norm}(\text{attn}(\text{Norm}(E)), \\ \mathcal{Z}^{'} &= \text{FC}(\mathcal{Z}; \theta_1), \\ \mathcal{Z}^{''} &= \text{FC}(\sigma(\mathcal{Z}^{'} + DS(\mathcal{Z}_i; \theta)); \theta_2), \end{aligned} \tag{7}$$

where $\text{DS}(\cdot; \theta_i)$ is the detail specific layer with parameters θ, implemented by a depth-wise convolution in practice, $\sigma(\cdot)$ represents GELU, $\text{Norm}(\cdot)$ represents the layer normalization, and $\text{attn}(\cdot)$ represents the SSA.

3.3 Crowd counting regression branch

Crowd counting regression branch has a regression head and a global average pooling operation. Specifically, we concat four features maps after each STB block followed by a regression head which is global average pooling operation. As shown in Fig. 1, we adopt global average pooling to regress predicted counts. Because the feature map of final stage may lose detail information of heads, we choose the outputs of the first three stages concatenated on the final output. Considering the convenience of feature fusion, we use a downsampling operation to transform feature maps $\{FM_i \mid i = 1, 2, 3, 4\}$ of different spatial sizes into the same size. Furthermore, we calculate the predicted counts of the image I ($\hat{C}_i$) following Eq. 8 as:

$$\hat{C}_i = \text{Pool2}D(\sum_{i=1}^{S} \text{DSA}(FM_i, \beta)), \tag{8}$$

where FM_i represents the feature map of the stage i. $DSA(\cdot, \beta)$ is the downsampling operation with downsampling rate β. $\text{Pool2}D(\cdot)$ stands for the global average pooling operation. The S is the number of stages. According to the Table 9, we found the S is 4 that has the best performance.

3.4 Crowd localization branch

As shown in Fig. 1, localization branch has three components which are the proposal generator, the proposal scorer and the proposal classifier, respectively.

Proposal generator. The proposal generator has a selective search module of Faster-RCNN [51]. The proposal generator uses selective search module to output 1000 proposals that correspond to different objects in the image.

Proposal scorer. The proposal scorer has a convolution layer and two deconvolution layers. We extract and fuse the feature maps of the first STB and the last STB. The fused feature maps are passed through a convolution layer and two deconvolution layers. The function of these two deconvolution layers is to recover the result to the same size as the original image so as to obtain a prediction score for each pixel. Furthermore, each proposed score is the mean of all pixel scores in the area covered by the proposal.

Proposal classifier. The proposal classifier uses these proposal scores to obtain the foreground and background regions of the image. We set the top 10% highest scoring overlapping proposals are labeled as foreground and the rest as background.

Finally, we acquire the foreground regions from proposal classifier and then use non-maximal suppression to select the center of the highest-scoring proposals as the final pseudo-point-level annotations in the foreground. The training of the weakly supervised crowd localization of CLFormer is performed in loops. In each loop, it alternates between generating pseudo-point-level annotations and updating the foreground region. Here, we set $\hat{C}_i$ as the predicted counts, and in each loop, we select the regions that intersect the top $10\% \times \hat{C}_i$ of highest scoring proposals based on the score map as the foreground. The rest are background. Finally, we see the all pseudo-point-level annotations as localization points. The process ends when the number of localization points in the image equals the predicted counts $\hat{C}_i$. And the CLFormer to increase the precise of crowd counting and crowd localization use the cross training. The details are as shown in Algorithm 2.

Algorithm 1 CLFormer Localization Training

Input: $r = 0.1$, $j = 0$, heatmap, $T_0 = 0$;
Output: the localization map
while $T_j < \hat{C}_i$ **do**:
  $j = j + 1$;
  Obtain the score maps;
  Generate the proposals $P_j = 1000$ from unlabeled region;
  Select top $r \times \hat{C}_i$ proposals($\hat{C}_i$ is the predicted counts);
  Obtain labeled and unlabeled regions for all images;
  Obtain pseudo point-level annotations T_j(part);
  $T_j = T_j + T_{j-1}$;
end while
Obtain all pseudo point-level annotations T;
Predicted localization maps = T;
return Predicted localization maps

Algorithm 2 CLFormer Cross Training

Input: the input images (I), CLFormer model ($model$), set_epoch=20000;
Output: predicted counts ($\hat{C}_i$), the localization map;
while epoch < set_epoch **do**:
  $\hat{C}_i$, heatmap = $model(I)$;
  put the heatmap and $\hat{C}_i$ into Algorithm 1;
end while
return Predicted counts, Predicted localization maps

3.5 Loss function

To train CLFormer, we minimize the difference between the predicted counts and the ground truth to directly increase the precise of crowd counting and the quality of pseudo-point-level annotations for crowd localization.

Specifically, we optimize our neural networks by two parts: crowd counting loss and crowd localization loss. The counting loss adopts $\mathcal{L}_1$ (Eq. 2) loss. The localization loss has four components: image level loss, pseudo-point level loss, spilt level loss and false positive loss. The loss function $\mathcal{L}$ is defined as:

$$\mathcal{L} = \mathcal{L}_1 + \mathcal{L}_{\text{image}} + \mathcal{L}_{\text{pseudo\_points}} + \mathcal{L}_{\text{split}} + \mathcal{L}_{\text{FP}}, \tag{9}$$

The image level loss can decrease the probability that the model labels any pixel as class c. The image level loss $\mathcal{L}_{image}$ is defined as:

$$\mathcal{L}_{\text{image}} = -\frac{1}{|C_e|}\sum_{c\in C_e}\log(S_{t_c c}) - \frac{1}{|C_{\neg e}|}\sum_{c\in C_{\neg e}}\log(S_{t_c c}), \tag{10}$$

where C_e represents people and $C_{\neg e}$ represents sets of classes that are not belonging to people. $S_{t_c c}$ represents the probability that pixel t_c belongs to category c. The t_c represents pixel i in the image which has the highest probability of belonging to category c.

The pseudo-point-level loss can increase the precision of labeling. The pseudo-point-level loss $\mathcal{L}_{\text{pseudo\_points}}$ is defined as:

$$\mathcal{L}_{\text{pseudo\_points}} = -\sum_{i\in I_s}\log(S_{iT_i}), \tag{11}$$

where T_i represents the previous label of pixel i.

The spilt level loss can reduce the probability of predicting blobs that have two or more points annotations. The spilt level loss $\mathcal{L}_{split}$ is defined as:

$$\mathcal{L}_{\text{split}} = -\sum_{i\in T_b}\alpha_i\log(S_{i0}), \tag{12}$$

where α_i represents the number of points annotations in the same places. S_{i0} represents the probability of the pixel i belongs to background.

The false positive loss can reduce the false label. And the false positive loss $\mathcal{L}_{\text{FP}}$ is defined as:

$$\mathcal{L}_{\text{FP}} = -\sum_{i\in B_{fp}}\log(S_{i0}), \tag{13}$$

where B_{fp} represents the set of points which belong to this prediction but not to the previous label.

4 Experiments

In this section, we first describe the implementation details and experiment setup. Then, we introduce the commonly used crowd counting datasets and compare our method with other state-of-the-art methods. Finally, we conduct ablation experiments to evaluate the effectiveness of each component from our method.

4.1 Implementation details

We apply the Adam to optimize our network, which is trained 20,000 epochs. We set the batch size as 4, the weight decay is set to 1e-4, the learning rate is set to 1e-5, in which after more than 300 training epochs, the learning rate is reduced to 0.1 times the initial learning rate. Furthermore, the weights pre-trained on ImageNet are used to initialize the Transformer-encoder. During training, the widely used data augmentation strategies are utilized, including random horizontal flipping and gray scaling. We resize all the images into the size of 384×384. For the localization task, we apply the fused feature maps to generate the score maps by adopting a 1×1 convolutional layer and two 4×4 deconvolutional layers with stride 2. Finally, the experiments are conducted under the PyTorch framework with a single NVIDIA GTX 2080Ti GPU.

4.2 Datasets

ShanghaiTech [18] is divided into two parts which are ShanghaiTechA and ShanghaiTechB. ShanghaiTechA contains 300 images for training and 182 images for testing. ShanghaiTechB contains 400 images for training and 316 images for testing.

UCF-QNRF [19] contains 1535 images that include one million annotations. The count range is very large which is from 49 to 12,865. Furthermore, its 1201 images for training and 334 images for testing.

NWPU-Crowd [15] is a dataset that is large-scale and challenging. It contains 5109 images, and 2,133,375 instances annotated elaborately. Meanwhile, the dataset is randomly divided into three parts: training dataset, validation dataset, and testing dataset, which contain 3,109,500, and 1500 images, respectively.

JHU-CROWD++ [20] contains 2722 training images, 500 validation images, and 1600 testing images, collected from diverse scenarios. The total number of people in each image ranges from 0 to 25,791.

4.3 Evaluation metric

We choose mean absolute error (MAE) and mean square error (MSE) to evaluate the counting performance:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|C_i - \hat{C}_i|, \tag{14}$$

$$MSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|\hat{C}_i - C_i|^2}, \tag{15}$$

where N is the number of testing images and $\hat{C}_i$ and C_i are the predicted and ground truth count of the i-th image, respectively. For the localization metric, we choose F1-measure, precision and recall to evaluate the localization performance.

4.4 Crowd counting comparisons

We conduct extensive experiments on four popular datasets. For each dataset, we divide the existing methods into fully supervised methods (based on point-level annotations [2,52, 53]) and weakly supervised methods (based on count-level annotations [8,11]), as shown in Tables 1, 2, 3, and 4.

Compared with the weakly supervised counting methods. Specifically, on the ShanghaiTechA, our method improves 10.20% in MAE and 12.50% in MSE compared to TransCrowd-GAP, the main reason for its improved counting performance is that the TransCrowd [11] network uses ViT [12] as its backbone and lacks the ability to obtain different scales information (Fig. 4); our method improves 6.00% in MAE and 1.70% in MSE compared to CCTrans [34], and the main reason for its improved counting performance is that CCTrans [34] uses a combination of dilated convolutions and vision Transformer which is limited by the size of the convolution kernel to capture more multi-scale and global information. Our method improves 13.20% in MAE and 20.60% in MSE compared to MATT [8], which benefits from Transformer models that can acquire global contextual information. Note that MATT [8] and CCTrans [34] still apply a small number of images, which contain density maps for training. Furthermore, we also compared these weakly supervised methods on QNRF, our method improves 8.19% in MAE and 9.71% in MSE compared to MATT [8] and improves 6.58% in MAE and 5.64% in MSE compared to TransCrowd-GAP [11], as shown in Table 2. On the JHU-CROWD++, our method improves 5.47% in MAE and 13.84% in MSE compared to TransCrowd-GAP [11], as shown in Table 3.

Compared with the fully supervised counting methods. We also compare CLFormer with the fully supervised counting methods, as shown in Tables 1 and 2. Our method improves 11.50% in MAE and 19.10% in MSE compared to CSRNet on ShanghaiTechA and improves 16.00% in MAE and 5.00% in MSE compared to CSRNet [53] on ShanghaiTechB [18], which benefits from Transformer models that can acquire global contextual information. Our method improves 3.60% in MAE and 7.90% in MSE compared to CSRNet [53] on ShanghaiTechA. Meanwhile, we also compared these fully supervised methods on QNRF, our method improves 26.77% in MAE and 18.88% in MSE compared to CSRNet [53], as shown in Table 2. For JHU-CROWD++, our method improves 17.58% in MAE and 17.63% in MSE compared to CSRNet [53], as shown in Table 3.

4.4.1 Cross-dataset evaluation

Finally, we conduct cross-dataset experiments on the UCF-QNRF [19], ShanghaiTechA [18] and ShanghaiTechB [18] datasets to explore the transferability of the proposed CLFormer. In the cross-dataset evaluation, models are trained on the source dataset and tested on the target dataset without further fine-tuning. The quantitative results are shown in Table 5. Although our method is a weakly supervised paradigm, we still achieve highly competitive performance compared with fully supervised methods [2,53].

Table 1 Quantitative comparisons (MAE and MSE) of different crowd counting methods on ShanghaiTech dataset

Methods	Publish	Training label		Part A		Part B	
		Localization	Crowd number	MAE ↓	MSE↓	MAE↓	MSE ↓
MCNN [2]	CVPR16	✓	✓	110.2	174.0	27.1	51.4
CSRNet [53]	CVPR18	✓	✓	68.4	116.0	10.6	16.0
BL [54]	ICCV19	✓	✓	62.8	101.8	8.0	13.1
S3 [55]	IJCAI21	✓	✓	57.1	97.3	8.3	12.6
UOT [21]	AAAI21	✓	✓	58.1	92.4	7.3	12.4
MATT [8]	PR21	×	✓	69.7	118.2	10.6	19.9
CCTrans [34]	arXiv21	×	✓	64.4	95.4	**7.0**	11.5
TransCrowd-token [11]	SCIS22	×	✓	69.7	118.2	10.6	19.9
TransCrowd-GAP [11]	SCIS22	×	✓	67.4	107.2	9.4	16.3
CLFormer (ours)	–	×	✓	**60.5**	**93.8**	8.9	**15.2**

Localization of training label represents point-level annotations. Crowd number represents count-level annotations. The bold fonts represent the best performance

Table 2 Quantitative comparisons (MAE and MSE) of different crowd counting methods on QNRF dataset

Methods	Publish	Training Label		QNRF	
		Localization	Crowd Number	MAE ↓	MSE ↓
MCNN [2]	CVPR16	✓	✓	277.0	426.0
CSRNet [53]	CVPR18	✓	✓	124.0	196.0
BL [54]	ICCV19	✓	✓	88.7	154.8
S3 [55]	IJCAI21	✓	✓	80.6	139.8
UOT [21]	AAAI21	✓	✓	83.3	142.3
MATT [8]	PR21	×	✓	98.9	176.1
CCTrans [34]	arXiv21	×	✓	92.1	**158.9**
TransCrowd-token [11]	SCIS22	×	✓	98.9	176.1
TransCrowd-GAP [11]	SCIS22	×	✓	97.2	168.5
CLFormer(ours)	–	×	✓	**90.8**	159.0

Bold fonts represent the best results

Table 3 Quantitative comparisons (MAE and MSE) of the crowd counting methods on JHU-CROWD++ dataset

Methods	Publish	Training Label		Testing set	
		Localization	Crowd Number	MAE ↓	MSE↓
MCNN [2]	CVPR16	✓	✓	188.9	483.4
CSRNet [53]	CVPR18	✓	✓	85.9	309.2
BL [54]	ICCV19	✓	✓	75.0	299.9
UOT [21]	AAAI21	✓	✓	60.5	252.7
S3 [55]	IJCAI21	✓	✓	**59.4**	**244.0**
MATT [8]	PR21	×	✓	86.2	417.9
TransCrowd-token [11]	SCIS22	×	✓	76.4	319.8
TransCrowd-GAP [11]	SCIS22	×	✓	74.9	295.6
CLFormer(ours)	–	×	✓	**70.8**	**254.7**

Bold fonts represent the best results

Table 4 Comparison (MAE and MSE) of the counting performance on the NWPU-Crowd datasets

Methods	Publish	Training label		Testing set	
		Localization	Crowd number	MAE ↓	MSE ↓
CSRNet [53]	CVPR18	✓	✓	121.3	387.8
BL [54]	ICCV19	✓	✓	105.4	454.2
UOT [21]	AAAI21	✓	✓	**83.5**	**346.9**
S3 [55]	IJCAI21	✓	✓	87.8	387.5
MATT [8]	PR21	×	✓	130.2	502.8
TransCrowd-token [11]	SCIS22	×	✓	124.6	390.1
TransCrowd-GAP [11]	SCIS22	×	✓	117.7	451.0
CLFormer(ours)	–	×	✓	**111.4**	**423.3**

Bold fonts represent the best results

4.4.2 Convergence curves comparisons

We further compared the convergence curves between the popular fully supervised method (i.e., CSRNet [53]), the popular weakly supervised method TransCrowd-GAP [11] and our proposed CLFormer, as shown in Fig. 5. Both TransCrowd-GAP [11] and CLFormer show smooth curves and fast convergence, but our proposed CLFormer converges faster than TransCrowd-GAP [11], while the curve of CSRNet is oscillatory.

4.5 Crowd localization comparisons

In this section, we evaluate our method on the NWPU-Crowd [15] dataset and the ShanghaiTechA [18] dataset.

Fig. 4 Examples of attention maps from TransCrowd-GAP [11] and our method CLFormer. The attention map of CLFormer generates more reasonable attention weights compared with the weight map of TransCrowd-GAP [11]

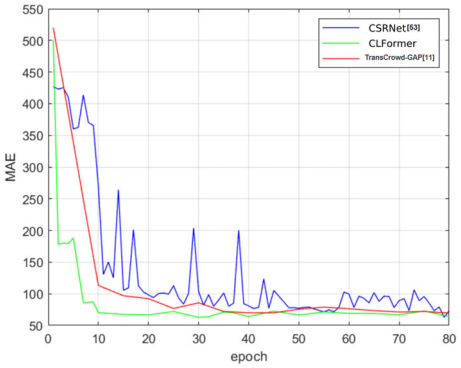

Fig. 5 Convergence curves of CSRNet [53], TransCrowd-GAP [11], our method CLFormer on ShanghaiTechA. Our method achieves the best counting performance and is fasting-converging

For the NWPU-Crowd [15] dataset, the results are shown in Table 6. Compared with fully supervised methods, our method improves 10.1% in F1-measure compared with VGG-16+GPR [17]. The main reason for the significant improvement in localization is that the Transformer-based network architecture can capture more global information than the convolutional network to better distinguish the foreground and background. Compared with weakly supervised methods, our method improves 0.65% in F1-measure compared with LOOC [16], which benefits from the Transformer architecture is not limited by the size of the receptive field compared to CNN models. In addition, we add our localization branch to the TransCrowd counting network [11] in order to verify the transferability of our localization branch. Our method improves 4% in F1-measure compared with TransCrowd [11], which benefits from the STB that can capture multi-scale information.

For the ShanghaiTechA [18] dataset, the results are shown in Table 7, Fig. 6; our method is compared with the popular fully supervised and weakly supervised methods. Compared with weakly supervised methods, our method improves 3% in F1-measure compared with TopK [50]. Due to the number of the ShanghaiTechA [18] dataset being much smaller than the number of the NWPU-Crowd [15] dataset, our method has a smaller number of dominant results on ShanghaiTechA. In addition, in order to test the transferability of the weakly

supervised localization method, we also evaluate the added localized TransCrowd [11] network on the ShanghaiTechA. Our method improves 2.1% in F1-measure compared with TransCrowd.

4.6 Comparison of inference-time

As shown in Table 8, we compare with two popular fully supervised counting methods, including BL [54] and CSRNet [53]; we also compare two popular weakly supervised counting methods, including TransCrowd-token [11] and TransCrowd-GAP [11]. The experiment is conducted on the NVIDIA GTX 2080Ti GPU. Even though the proposed CLFormer has more run-time than other methods, it still achieve outstanding performance with only half of parameters. Because our method selectively merges tokens to represent larger object features while keeping certain tokens to preserve fine-grained features. This merging scheme enables the self-attention to learn relationships between objects with different sizes and simultaneously reduces the token numbers and the computational cost. Additionally, we can observe that the FPS of VGG19-based BL [54] outperforms the VGG16-based CSRNet [53], mainly because the BL generates a small-resolution density map (1/16 of the input image). This phenomenon further demonstrates the influence of feature resolution on run-time.

4.7 Ablation studies

Our proposed CLFormer contains four STBs. For the counting task, we examine the counting performance by changing the number of fused feature maps, as shown in Table 9. According to experiments, the counting performance improves significantly as the number of STBs increases, the best counting performance is achieved when the STB is 4. However, when adding another STB with the extraction and fusion, we find that the MAE value increases. Thus, we choose the number of STBs is 4 as the part of our network.

For the localization task, we test the accuracy of the localization by changing the position of the fused feature map, as shown in Table 10. Due to the strong semantic information of the feature map output after the last STB, which can extract the foreground from the complex background, we retain the feature map output after the last STB. Meanwhile, to compensate for the detailed information loss, we will fuse the feature map output from one of the previous STBs. According to the data in Table 10 we find that fusing the feature maps from the first STB output achieves the best localization performance.

In addition to this, we increased the number of fused feature maps based on the first and last STB output in order to see whether increasing the number of feature maps would improve the localization performance, the results are shown

Table 5 The transferability of different methods under cross-dataset evaluation

Methods	Backbone	Publish	Training label		PartA→ partB		PartB→ partA		QNRF→ partA		QNRF→ partB	
			Localization	Crowd number	MAE ↓	MSE ↓	MAE ↓	MSE ↓	MAE ↓	MSE↓	MAE ↓	MSE ↓
MCNN [2]	VGG16	CVPR16	✓	✓	86.3	140.6	223.5	360.1	–	–	–	–
BL [54]	VGG19	ICCV19	✓	✓	–	–	–	–	69.8	124.0	15.4	25.5
TransCrowd-token [11]	vision Transformer	SCIS22	×	✓	20.1	50.3	150.6	270.3	80.1	128.6	15.3	24.7
TransCrowd-GAP [11]	vision Transformer	SCIS22	×	✓	18.9	31.1	142.6	257.3	78.6	122.7	13.5	22.9
CLFormer(ours)	Shunted Transformer	–	×	✓	**17.4**	**28.6**	**114.5**	**210.5**	**65.8**	**108.0**	**13.0**	**21.7**

Bold fonts represent the best results

Fig. 6 The visualization of the localization of our method on ShanghaiTechA. The red circle represents the predicted localization, and the green circle represents the ground truth

Table 6 Quantitative comparisons (F1-measure, precision and recall) of different crowd localization methods on NWPU-Crowd dataset [11]

Methods	Backbone	Training label				Overall		
		Points	Box	Counts	Pseudo tag	F1-Measure(%)↑	Precision(%)↑	Recall(%)↑
Faster-RCNN [51]	ResNet101	✓	✓	✓	×	6.8	90.0	3.5
VGG-16+GPR [17]	VGG16	✓	✓	✓	×	56.3	61.0	52.2
RAZ_LOC [56]	VGG16	✓	✓	✓	×	**62.5**	**69.2**	56.9
TopK [50]	ResNet50	×	×	✓	✓	44.8	64.5	34.3
LOOC [16]	VGG16	×	×	✓	✓	61.6	66.8	57.2
TransCrowd-GAP(*) [11]	Vision transformer	×	×	✓	✓	59.6	65.4	54.8
CLFormer (ours)	Shunted Transformer	×	×	✓	✓	62.0	67.4	**57.5**

*Represents the proposed localization branch that is used to TransCrowd-GAP
Bold fonts represent the best results

Table 7 Quantitative comparisons (F1-measure, precision and recall) of different crowd localization methods on ShanghaiTechA

Methods	Backbone	Training label				Overall		
		Points	Box	Counts	Pseudo tag	F1-Measure(%)↑	Precision(%) ↑	Recall(%)↑
Faster-RCNN [51]	ResNet101	✓	✓	✓	✓	18.3	**100**	10.1
VGG-16+GPR [17]	VGG16	✓	✓	✓	✓	58.3	63.2	54.1
RAZ_LOC [56]	VGG16	✓	✓	✓	✓	**65.7**	72.1	60.3
TopK [50]	ResNet50	×	×	✓	✓	46.6	65.2	36.1
LOOC [16]	VGG16	×	×	✓	✓	**60.7**	**63.2**	**58.4**
TransCrowd-GAP(*) [11]	vision Transformer	×	×	✓	✓	47.0	41.6	54.0
CLFormer (ours)	Shunted Transformer	×	×	✓	✓	48.0	42.9	54.3

Bold fonts represent the best results

Table 8 Computational resources comparisons of different methods

Methods	Resolution	Parameters↓	Backbone	FPS ↑
CSRNet [53]	384×384	**16.2 M**	VGG16	21.67
BL [54]	384×384	21.6 M	VGG19	**45.66**
TransCrowd-token [11]	384×384	86.8 M	Vision transformer	46.41
TransCrowd-GAP [11]	384×384	90.4 M	Vision transformer	**46.73**
CLFormer(ours)	384×384	**49.1 M**	Shunted transformer	44.94

Bold fonts represent the best results

Table 9 Quantitative comparisons (MAE and MSE) of CLFormer with different number of stages

The number of STB	Metric	
	MAE↓	MSE↓
2	112.7	181.8
3	99.8	164.4
4	**60.5**	**93.8**
5	62.5	95.6

Bold fonts represent the best results

Table 10 Quantitative comparisons (F1-measure, precision and recall) of different position

Position of feature map	Metric		
	F1-measure(%)↑	Precision(%)↑	Recall(%)↑
1	**62.0**	**67.4**	57.5
2	37.6	24.6	**79.8**
3	49.3	37.4	72.4

Bold fonts represent the best results

Table 11 Quantitative comparisons (F1-measure, precision and recall) of adding different number of stages

Add different feature maps	Metric		
	F1-measure(%)↑	Precision(%)↑	Recall(%)↑
4+1	**62.0**	**67.4**	57.5
4+1+2	62.0	67.3	57.5
4+1+3	61.6	66.1	**57.7**
4+1+2+3	62.0	66.5	57.5

Bold fonts represent the best results

in Table 11. According to Table 11, we found that increasing the number of feature maps does not significantly improve the localization performance. Thus, we chose to use the results of the fusion of the first STB with the last STB for the localization task.

5 Conclusion

In this paper, we propose a CLFormer for multi-scale crowd counting and crowd localization. The CLFormer has two branches that are the regression branch and localization branch, respectively. The CLFormer adopts the Shunted Transformer as the backbone which can capture the multi-scale features. The regression branch leverages the global average pooling to obtain efficient multi-scale crowd features information for final predicted counts estimation. The localization branch has three components that are proposal generator, proposal scorer and proposal classifier. We generate proposals from the proposal generator and utilize the proposal scorer to rescore these proposals and then filter these proposals and obtain some top high-scoring proposals to generate the pseudo-point-level annotations. When the number of pseudo-point-level annotations equals the predicted counts from regression branch, we watch these pseudo-point-level annotations as the localization maps. CLFormer is evaluated on four challenging crowd counting datasets and achieves superior results compared with other state-of-the-art methods and both quantitative and qualitative results, demonstrating the effectiveness of our method. We only consider the image crowd counting and do not explore the video crowd counting, which is more suitable for real-world application. In future work, we will extend our work for weakly supervised video crowd counting task.

Data availability The datasets generated or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The author declare that they have no conflict of interest.

References

1. Tripathi, G., Singh, K., Vishwakarma, D.K.: Convolutional neural networks for crowd behaviour analysis: a survey. Vis. Comput. **35**, 753–776 (2019)
2. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 589–597 (2016)
3. Wang, S., Lu, Y., Zhou, T., Di, H., Lu, L., Zhang, L.: Sclnet: spatial context learning network for congested crowd counting. Neurocomputing **404**, 227–239 (2020)
4. Xie, Y., Lu, Y., Wang, S.: Rsanet: deep recurrent scale-aware network for crowd counting. In: 2020 IEEE International Conference on Image Processing (ICIP), pp. 1531–1535 (2020)
5. Duan, Z., Wang, S., Di, H., Deng, J.: Distillation remote sensing object counting via multi-scale context feature aggregation. IEEE Trans. Geosci. Remote Sens. **60**, 1–12 (2021)
6. Chen, X., Yu, X., Di, H., Wang, S.: Sa-internet: scale-aware interaction network for joint crowd counting and localization. In: Pattern Recognition and Computer Vision, pp. 203–215. Springer, Heidelberg (2021)
7. Zhang, L., Yan, L., Zhang, M., Lu, J.: T2 cnn: a novel method for crowd counting via two-task convolutional neural network. Vis. Comput. **39**(1), 73–85 (2023)
8. Lei, Y., Liu, Y., Zhang, P., Liu, L.: Towards using count-level weak supervision for crowd counting. Pattern Recognit. **109**, 107616 (2021)
9. Yang, Y., Li, G., Wu, Z., Su, L., Huang, Q., Sebe, N.: Weakly-supervised crowd counting learns from sorting rather than locations. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 1–17 (2020)

10. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 6000–6010 (2017)
11. Liang, D., Chen, X., Xu, W., Zhou, Y., Bai, X.: Transcrowd: weakly-supervised crowd counting with transformers. Sci. China Inf. Sci. **65**(6), 1–14 (2022)
12. Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. International Conference on Learning Representations (2021)
13. Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 568–578 (2021)
14. Ren, S., Zhou, D., He, S., Feng, J., Wang, X.: Shunted self-attention via multi-scale token aggregation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10853–10862 (2022)
15. Wang, Q., Gao, J., Lin, W., Li, X.: Nwpu-crowd: a large-scale benchmark for crowd counting and localization. IEEE Trans. Pattern Anal. Mach. Intell. **43**(6), 2141–2149 (2020)
16. Laradji, I.H., Pardinas, R., Rodriguez, P., Vazquez, D.: Looc: localize overlapping objects with count supervision. In: International Conference on Image Processing (ICIP), pp. 2316–2320 (2020)
17. Gao, J., Han, T., Wang, Q., Yuan, Y.: Domain-adaptive crowd counting via inter-domain features segregation and gaussian-prior reconstruction. arXiv preprint arXiv:1912.03677
18. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 589–597 (2016)
19. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 532–546 (2018)
20. Sindagi, V., Yasarla, R., Patel, V.M.: JHU-CROWD++: large-scale crowd counting dataset and a benchmark method. IEEE Trans. Pattern Anal. Mach. Intell. **44**(5), 2594–2609 (2020)
21. Ma, Z., Wei, X., Hong, X., Lin, H., Qiu, Y., Gong, Y.: Learning to count via unbalanced optimal transport. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 2319–2327 (2021)
22. Li, Z., Lu, S., Dong, Y., Guo, J.: Msffa: a multi-scale feature fusion and attention mechanism network for crowd counting. Vis. Comput., 1–12 (2022)
23. Khan, S.D., Basalamah, S.: Scale and density invariant head detection deep model for crowd counting in pedestrian crowds. Vis. Comput. **37**(8), 2127–2137 (2021)
24. Wang, Y., Ma, Z., Wei, X., Zheng, S., Wang, Y., Hong, X.: Eccnas: efficient crowd counting neural architecture search. ACM Trans. Multim. Comput. Commun. Appl. **18**(1s), 1–19 (2022)
25. Shi, M., Yang, Z., Xu, C., Chen, Q.: Revisiting perspective information for efficient crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7279–7288 (2019)
26. Yang, Y., Li, G., Wu, Z., Su, L., Huang, Q., Sebe, N.: Reverse perspective network for perspective-aware object counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4374–4383 (2020)
27. Zhou, T., Li, J., Wang, S., Tao, R., Shen, J.: Matnet: motion-attentive transition network for zero-shot video object segmentation. IEEE Trans. Image Process. **29**, 8326–8338 (2020)
28. Wang, S., Zhou, T., Lu, Y., Di, H.: Contextual transformation network for lightweight remote-sensing image super-resolution. IEEE Trans. Geosci. Remote Sens. **60**, 1–13 (2021)
29. Zhou, T., Li, L., Li, X., Feng, C.-M., Li, J., Shao, L.: Group-wise learning for weakly supervised semantic segmentation. IEEE Trans. Image Process. **31**, 799–811 (2021)
30. Wang, S., Zhou, T., Lu, Y., Di, H.: Detail-preserving transformer for light field image super-resolution. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, pp. 2522–2530 (2022)
31. Zhang, M., Li, J., Zhou, T.: Multi-granular semantic mining for weakly supervised semantic segmentation. In: Proceedings of the 30th ACM International Conference on Multimedia, pp. 6019–6028 (2022)
32. Lai, Q., Zhou, T., Khan, S., Sun, H., Shen, J., Shao, L.: Weakly supervised visual saliency prediction. IEEE Trans. Image Process. **31**, 3111–3124 (2022)
33. Zhou, T., Li, L., Bredell, G., Li, J., Unkelbach, J., Konukoglu, E.: Volumetric memory network for interactive medical image segmentation. Med. Image Anal. **83**, 102599 (2023)
34. Tian, Y., Chu, X., Wang, H.: Cctrans: Simplifying and improving crowd counting with transformer. arXiv preprint arXiv:2109.14483 (2021)
35. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: revisiting the design of spatial attention in vision transformers. Adv. Neural Inf. Process. Syst. **34**, 9355–9366 (2021)
36. Li, B., Zhang, Y., Xu, H., Yin, B.: Ccst: crowd counting with swin transformer. Vis. Comput., 1–12 (2022)
37. Lin, H., Ma, Z., Ji, R., Wang, Y., Hong, X.: Boosting crowd counting via multifaceted attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19628–19637 (2022)
38. Sajid, U., Chen, X., Sajid, H., Kim, T., Wang, G.: Audio-visual transformer based crowd counting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 2249–2259 (2021)
39. Liu, Y., Ren, S., Chai, L., Wu, H., Xu, D., Qin, J., He, S.: Reducing spatial labeling redundancy for active semi-supervised crowd counting. IEEE Trans. Pattern Anal. Mach. Intell. (2022)
40. Liu, N., Long, Y., Zou, C., Niu, Q., Pan, L., Wu, H.: Adcrowdnet: an attention-injective deformable convolutional network for crowd understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3225–3234 (2019)
41. Sam, D.B., Peri, S.V., Sundararaman, M.N., Kamath, A., Babu, R.V.: Locate, size, and count: accurately resolving people in dense crowds via detection. IEEE Trans. Pattern Anal. Mach. Intell. **43**(8), 2739–2751 (2020)
42. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European Conference on Computer Vision, pp. 213–229. Springer, Heidelberg (2020)
43. Liu, Y., Shi, M., Zhao, Q., Wang, X.: Point in, box out: Beyond counting persons in crowds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6469–6478 (2019)
44. Idrees, H., Tayyab, M., Athrey, K., Zhang, D., Al-Maadeed, S., Rajpoot, N., Shah, M.: Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 532–546 (2018)
45. Liang, D., Xu, W., Zhu, Y., Zhou, Y.: Focal inverse distance transform maps for crowd localization. IEEE Trans. Multim. (2022)
46. Xu, C., Liang, D., Xu, Y., Bai, S., Zhan, W., Bai, X., Tomizuka, M.: Autoscale: learning to scale for crowd counting. Int. J. Comput. Vis. **130**(2), 405–434 (2022)

47. Song, Q., Wang, C., Jiang, Z., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Wu, Y.: Rethinking counting and localization in crowds: A purely point-based framework. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 3365–3374 (2021)
48. Zhang, D., Han, J., Cheng, G., Yang, M.-H.: Weakly supervised object localization and detection: a survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(9), 5866–5885 (2022)
49. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
50. Topkaya, I.S., Erdogan, H., Porikli, F.: Counting people by clustering person detector outputs. In: 2014 11th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 313–318 (2014)
51. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**(6), 1137–1149 (2016)
52. Babu Sam, D., Surya, S., Venkatesh Babu, R.: Switching convolutional neural network for crowd counting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5744–5752 (2017)
53. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1091–1100 (2018)
54. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 6142–6151 (2019)
55. Lin, H., Hong, X., Ma, Z., Wei, X., Qiu, Y., Wang, Y., Gong, Y.: Direct measure matching for crowd counting. arXiv preprint arXiv:2107.01558 (2021)
56. Liu, C., Weng, X., Mu, Y.: Recurrent attentive zooming for joint crowd counting and precise localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1217–1226 (2019)

Mingfang Deng was born in Xinjiang, China, and graduated from China University of Petroleum (Beijing) in 2020 with a B.S. degree. She is currently pursuing a master's degree in Control Science and Engineering at the Electrical and Electronic Engineering, Shanghai Institute of Technology. Her main research interests are deep learning and visual language navigation.

Huailin Zhao received his PhD from Oita University, Japan, in 2008. He is a professor in the School of Electrical and Electronic Engineering, Shanghai Institute of Technology, China. His main research interests are robotics, multi-agent system and artificial intelligence. He is the member of both IEEE and Sigma Xi.

Ming Gao was born in Henan, China, and graduated from Henan University of Science and Technology in 2020 with a B.S. degree. He is currently pursuing a master's degree in Control Science and Engineering at the Electrical and Electronic Engineering, Shanghai Institute of Technology. His main research interests are deep learning and visual language navigation.