SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •
Special Focus on Deep Learning for Computer Vision

June 2022, Vol. 65 160104:1–160104:14
https://doi.org/10.1007/s11432-021-3445-y

TransCrowd: weakly-supervised crowd counting with transformers

Dingkang LIANG[1], Xiwu CHEN[2], Wei XU[3], Yu ZHOU[2] & Xiang BAI[1*]

[1]*School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan* 430074*, China;*
[2]*School of Electronic Information and Communication, Huazhong University of Science and Technology, Wuhan* 430074*, China;*
[3]*School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing* 100876*, China*

Received 12 July 2021/Revised 19 December 2021/Accepted 27 January 2022/Published online 26 April 2022

Abstract The mainstream crowd counting methods usually utilize the convolution neural network (CNN) to regress a density map, requiring point-level annotations. However, annotating each person with a point is an expensive and laborious process. During the testing phase, the point-level annotations are not considered to evaluate the counting accuracy, which means the point-level annotations are redundant. Hence, it is desirable to develop weakly-supervised counting methods that just rely on count-level annotations, a more economical way of labeling. Current weakly-supervised counting methods adopt the CNN to regress a total count of the crowd by an image-to-count paradigm. However, having limited receptive fields for context modeling is an intrinsic limitation of these weakly-supervised CNN-based methods. These methods thus cannot achieve satisfactory performance, with limited applications in the real world. The transformer is a popular sequence-to-sequence prediction model in natural language processing (NLP), which contains a global receptive field. In this paper, we propose TransCrowd, which reformulates the weakly-supervised crowd counting problem from the perspective of sequence-to-count based on transformers. We observe that the proposed TransCrowd can effectively extract the semantic crowd information by using the self-attention mechanism of transformer. To the best of our knowledge, this is the first work to adopt a pure transformer for crowd counting research. Experiments on five benchmark datasets demonstrate that the proposed TransCrowd achieves superior performance compared with all the weakly-supervised CNN-based counting methods and gains highly competitive counting performance compared with some popular fully-supervised counting methods.

Keywords crowd counting, visual transformer, weakly supervised, crowd analysis, transformer

Citation Liang D K, Chen X W, Xu W, et al. TransCrowd: weakly-supervised crowd counting with transformers. Sci China Inf Sci, 2022, 65(6): 160104, https://doi.org/10.1007/s11432-021-3445-y

1 Introduction

Crowd counting is a hot topic in the computer vision community, which plays an essential role in video surveillance, public safety, and crowd analysis. Typical crowd counting methods [1–4] usually utilize the convolution neural network (CNN) to regress a density map, which has achieved significant progress recently. A standard regressor consists of an encoder and a decoder: the encoder extracts the high-level feature information, and the decoder is designed for pixel-level regression based on the extracted feature.

However, these density-map regression-based methods [1–3, 5] still have some drawbacks. (1) They apply the point-level annotations to generate ground-truth density maps, which are usually expensive cost. Actually, some methods [6–9] discover that we can collect a new crowd dataset by using a more economical strategy, such as mobile crowd-sensing [7] technology or GPS-less [8] energy-efficient sensing scheduling. For a given crowd scene with different viewpoints and the same total count (such as auditoria, classroom), if we know the total counts of one viewpoint, then the total counts of other viewpoints are known. Besides, we can obtain the crowd number at a glance for some sparse crowd scenes. (2) The annotated point label will not be taken to evaluate the counting performance, meaning that the point

* Corresponding author (email: xbai@hust.edu.cn)

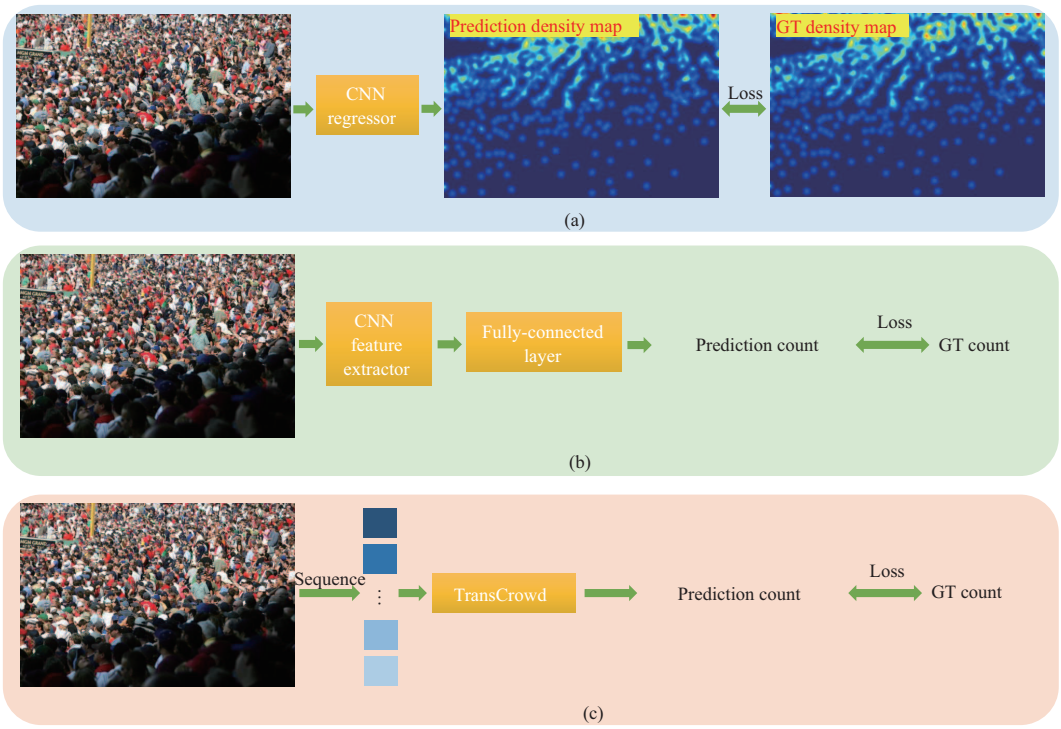

Figure 1 (Color online) (a) Traditional fully-supervised CNN-based method. All the training images are labeled with point-level annotations. (b) Weakly-supervised CNN-based method from an image-to-count perspective, only relying on the annotated total count of the crowd. (c) The proposed TransCrowd, a weakly-supervised method, reformulates the counting problem from the sequence-to-count perspective.

label is redundant to some extent. Thus, point-level annotations are not absolutely necessary for the crowd counting task.

Based on the above observations, it is desirable to develop the count-level crowd counting method. Following previous studies [6, 9], we call the methods which rely on the point-level annotations are fully-supervised paradigms, and the methods which only rely on count-level are weakly-supervised paradigms. The fully-supervised methods first utilize the point annotation to generate the ground-truth density map and then elaborately design a regressor to generate a prediction density map and finally apply the $L2$ loss to measure the difference between the prediction and the ground truth, as shown in Figure 1(a). The existing weakly-supervised methods usually regress the total count of crowd images directly, which is from the image-to-count perspective, as shown in Figure 1(b).

Recently, the transformer [10], a popular language model proposed by Google, has been explored in many vision tasks, such as classification [11], detection [12, 13], and segmentation [14]. Unlike the CNN, which utilizes a limited receptive field, the transformer [10] provides the global receptive field, showing excellent advantages over pure CNN architectures. In this paper, we propose TransCrowd, which is the first to explore the transformer into the weakly-supervised crowd counting task, establishing the perspective of sequence-to-count prediction, as illustrated in Figure 1(c).

Only a few methods are proposed with the consideration of reducing the annotations burden (e.g., semi, weakly-supervised). L2R [15] facilitates the counting task by ranking the image patch. Wang et al. [16] introduced a synthetic crowd dataset named GCC, and the model is pre-trained on the GCC dataset and then fine-tuned on real data. One of the most relevant studies for our method is [6], which proposes a soft-label network to facilitate the counting task, directly regressing the crowd number without the supervision of location labels. However, Ref. [6] is a CNN-based method, which has a limited respective field. The transformer has a global receptive field, which effectively solves the limited respective field problem of CNN-based methods once and for all. It means that the transformer architecture is more suitable for the weakly-supervised counting task since the task aims to directly predict a total count from the whole image and rely on the global perspective.

In this paper, we introduce two types of TransCrowd, named TransCrowd-Token and TransCrowd-GAP, respectively. TransCrowd-Token utilizes an extra learnable token to represent the count.

TransCrowd-GAP adopts the global average pooling (GAP) over all items in the output sequence of transformer-encoder to obtain the pooled visual tokens. The regression tokens or pooled visual tokens are then fed into the regression head to generate the prediction count. We empirically find that the TransCrowd-GAP can obtain more reasonable attention weight, achieve a higher count accuracy, and present fast-converging compared with TransCrowd-Token.

In summary, this article contributes to the following.

(1) TransCrowd is the first pure transformer-based crowd counting framework. We reformulate the counting problem from a sequence-to-count perspective and propose a weakly-supervised counting method, which only utilizes the count-level annotations without the point-level information in the training phase.

(2) We provide two different types of TransCrowd, named TransCrowd-Token and TransCrowd-GAP, respectively. We observe that the TransCrowd-GAP can generate a more reasonable attention weight and reports faster converging and higher counting performance than TransCrowd-Token.

(3) Extensive experiments demonstrate that the proposed method achieves state-of-the-art counting performance compared with the weakly-supervised methods. Additionally, our method has a highly competitive counting performance compared with the fully-supervised counting methods.

2 Related work

2.1 CNN-based crowd counting

The CNN-based crowd counting methods can be categorized into localization-based methods and regression-based methods. The localization-based methods [17, 18] usually learn to predict bounding boxes for each human, relying on box-level annotations. Recently, some methods [19–23] try to utilize the pseudo bounding boxes based on point-level annotations or design a suitable map to realize counting and localization tasks. However, these localization-based methods usually report unsatisfactory counting performance. The mainstream of crowd counting is the density map CNN-based crowd counting methods [1, 2, 24–28], whose integration of the density map gives the total count of a crowd image. Due to the commonly heavy occlusion that exists in crowd images, multi-scale architecture is developed. Specifically, MCNN [1] utilizes multi-size filters to extract different scale feature information. Sindagi et al. [29] captured the multi-scale information by the proposed contextual pyramid CNN. TEDNet [30] assembles multiple encoding-decoding paths hierarchically to generate a high-quality density map for accurate crowd counting. Method in [31] proposes a scale-aware probabilistic model to handle large scale variations through the density pyramid network (DPN), and each level of DPN copes with a given scale range. Using the perspective information to diminish the scale variations is effective [32–34]. PACNN [32] proposes a novel generating ground-truth perspective maps strategy and predicts both the perspective maps and density maps at the testing phase. Yang et al. [34] proposed a reverse perspective network to estimate the perspective factor of the input image and then warped the image. Appropriate measure matching can help to improve the counting performance. S3 [35] proposes a novel measure matching based on Sinkhorn divergence, avoiding generating the density maps. UOT [36] uses unbalanced optimal transport (UOT) distance to quantify the discrepancy between two measures, outputting sharper density maps.

The attention-based mechanism is another useful technique adopted by many methods [24, 27, 37]. ADCrowdNet [37] generates an attention map for the crowd images via a network called attention map generate (AMG). Jiang et al. [27] proposed a density attention network to generate attention masks concerning regions of different density levels. Zhang et al. [24] proposed a relation attention network (RANet) that utilizes local self-attention and global self-attention to capture long-range dependencies. It is noteworthy that RANet [24] is actually a non-local/self-attention mechanism based on CNN instead of pure transformers, and we utilize a pure transformer without convolution layers.

2.2 Weakly-supervised crowd counting

Only a few methods focus on counting with a lack of labeled data. L2R [15] proposes a learning-to-rank framework based on an extra collected crowd dataset. Wang et al. [16] introduced a synthetic crowd scene for the pre-trained model. However, these two methods still rely on point-level annotations, which are fully-supervised instead of weakly-supervised paradigms.

The traditional method [38] relies on hand-crafted features, such as GLCM and edge orientations, which are turned to be sub-optimal for this weakly-supervised counting task. MATT [9] learns a model from a small amount of point-level annotations (fully-supervised) and a large amount of count-level annotations (weakly-supervised). The method in [39] proposes a weakly-supervised solution based on the Gaussian process for crowd density estimation. Shang et al. [40] simultaneously predicted the global count and local count. Wang et al. [41] directly regressed the global count, and some negative samples are fed into the network to boost the robustness. Similarly, Yang et al. [6] also directly mapped the images to the crowd numbers without the location supervision based on the proposed soft-label sorting network.

However, the counting performance of these count-level weakly-supervised counting methods still does not achieve comparable results to the fully-supervised counting methods, existing massive degradation, limiting the application of weakly-supervised methods in the real world. Different from the previous studies, the proposed TransCrowd utilizes the transformer architecture to directly regress the crowd number, which formulates the counting problem as the sequence-to-count paradigm and achieves comparable counting performance compared with the popular fully-supervised methods.

2.3 Visual transformer

The transformers [10], dominating the natural language modeling [42,43], utilize the self-attention mechanism to capture the global dependencies between input and output. Recently, many studies [11,12,14, 44,45] attempt to apply the transformer to the vision task. Specifically, DETR [12] firstly utilizes a CNN backbone to extract the visual features, followed by the transformer blocks for the box regression and classification. ViT [11] is the first which directly applies transformer-encoder [10] to sequences of image patches to realize the classification task. SETR [14] regards semantic segmentation from a sequence-to-sequence perspective with transformers. IPT [45] develops a pre-trained model for image processing (low-level task) using the transformer architecture.

To the best of our knowledge, we are the first to explore the pure transformer [10] to the counting task.

3 Our method

The overview of our method consists of the sequence (tokens) of the image, a transformer-encoder, and a naive regression head, as shown in Figure 2(a). Specifically, the input image is first transformed into fixed-size patches and then flatten to a sequence of vectors. The sequence is fed into the transformer-encoder, followed by a naive regression head to generate the prediction count.

3.1 Image to sequence

In general, the transformer adopts a $1D$ sequence of feature embeddings $Z \in \mathbb{R}^{N\times D}$ as input, where N is the length of the sequence and the D means the input channel size. Thus, the first step of TransCrowd is to transform the input image I into a sequence of 2D flattened patches. Specifically, given an RGB image[1)] $I \in \mathbb{R}^{H\times W\times 3}$, we reshape I into a grid of N patches, resulting in $\{x_p^i \in \mathbb{R}^{K^2\times 3} | i = 1,\ldots,N\}$, where $N = \frac{H}{K} \times \frac{W}{K}$ and K is the pre-defined patch size.

3.2 Patch embedding

Next, we need to map the x into a latent D-dimensional embedding feature by a learnable projection, since the transformer uses constant latent vector size D through all of its layers, defined as

$$e = [e_1; e_2; \cdots; e_N] = [x_p^1 E; x_p^2 E; \cdots; x_p^N E], \qquad E \in \mathbb{R}^{(K^2\times 3)\times D}, \tag{1}$$

where E is a learnable matrix, and $e \in \mathbb{R}^{N\times D}$ is the mapped features. Thus, we add a specific position embedding $\{p_i \in \mathbb{R}^D | i = 1,\ldots,N\}$ into e, maintaining position information, defined as

$$Z_0 = [Z_0^1; Z_0^2; \cdots; Z_0^N] = [e_1 + p_1; e_2 + p_2; \cdots; e_N + p_N], \tag{2}$$

where Z_0 is the input of the first transformer layer.

1) H, W, 3 indicate the spatial height, width, and channel number, respectively.

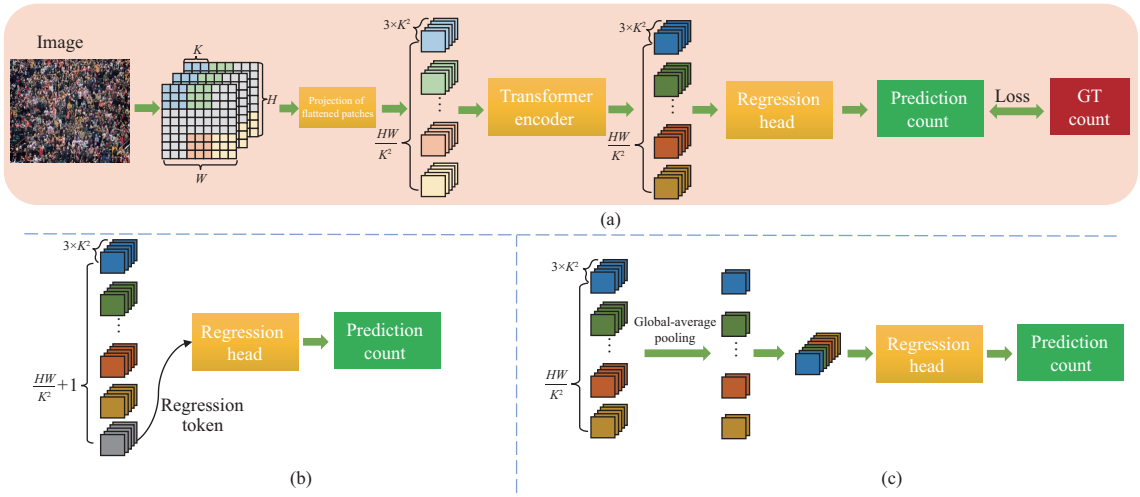

Figure 2 (Color online) (a) The pipeline of TransCrowd. The input image is split into fixed-size patches, each of which is linearly embedded with position embeddings. Then, the feature embedding sequence is fed into the transformer-encoder, followed by a regression head to generate the prediction count. (b) We utilize an extra token to represent the crowd count, similar to the class token in Bert [42] and ViT [11]. (c) A global average pooling is adopted to pool the output visual tokens of the transformer-encoder.

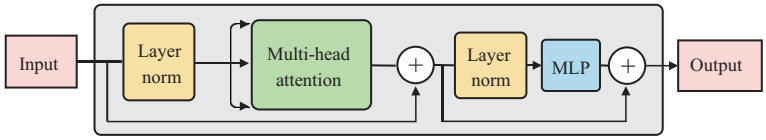

Figure 3 (Color online) A standard transformer layer consists of multi-head attention and MLP blocks. Meanwhile, the layer normalization (LN) and residual connections are employed.

3.3 Transformer-encoder

We only adopt the transformer encoder [10], without the decoder, similar to ViT [11]. Specifically, the encoder contains L layers of multi-head self-attention (MSA) and multilayer perceptron (MLP) blocks. For each layer l, layer normalization (LN) and residual connections are employed. A stand transformer layer is shown in Figure 3, and the output can be written as follows:

$$Z'_l = \mathrm{MSA}(\mathrm{LN}(Z_{l-1})) + Z_{l-1}, \tag{3}$$

$$Z_l = \mathrm{MLP}(\mathrm{LN}(Z'_l)) + Z'_l, \tag{4}$$

where Z_l is the output of layer l. Here, the MLP contains two linear layers with a GELU [46] activation function. In particular, the first linear layer of MLP expands the feature embedding's dimension from D to $4D$, while the second layer shrinks the dimension from $4D$ to D.

MSA is an extension with m independent self-attention (SA) modules: $\mathrm{MSA}(Z_{l-1}) = [\mathrm{SA}_1(Z_{l-1}); \mathrm{SA}_2(Z_{l-1}); \cdots; \mathrm{SA}_m(Z_{l-1})]W_O$, where $W_O \in \mathbb{R}^{D\times D}$ is a re-projection matrix. At each independent SA, the input consists of query (Q), key (K), and value (V), which are computed from Z^{l-1}:

$$Q = Z_{l-1}W_Q, \quad K = Z_{l-1}W_K, \quad V = Z_{l-1}W_V, \tag{5}$$

$$\mathrm{SA}(Z_l) = \mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{D}}\right)V, \tag{6}$$

where $W_Q/W_K/W_V \in \mathbb{R}^{D\times\frac{D}{m}}$ are three learnable matrices. The softmax function is applied over each row of the input matrix and $\sqrt{D}$ provides appropriate normalization.

3.4 The input of regression head

We introduce two different inputs for the regression heads to evaluate the effectiveness of TransCrowd. The goal of the regression head is to generate the prediction count instead of the density map. We briefly describe the two types of input.

(1) Regression token. Similar to the class token in Bert [42] and ViT [11], we prepend a learnable embedding named regression token to the input sequence Z_0, as shown in Figure 2(b) [42]. This architecture forces the self-attention to spread information between the patch tokens and the regression token, making the regression token contain overall semantic crowd information. The regression head is implemented by MLP containing two linear layers. We refer to the TransCrowd with the extra regression token as TransCrowd-Token.

(2) Global average pooling. We apply the global average pooling (GAP) to shrink the sequence length, as shown in Figure 2(c). Similar to TransCrowd-Token, two linear layers are used for the regression head. We refer to the TransCrowd with global average pooling as TransCrowd-GAP. The global average pooling can effectively maintain the useful semantic crowd information in patch tokens. We find that using pooled visual tokens will generate richer discriminative semantic crowd patterns and achieve better counting performance than using the extra regression token, the detailed discussion listed in Section 6.

We utilize L_1 loss to measure the difference between the prediction and the ground truth:

$$L_1 = \frac{1}{M}\sum_{i=1}^{M}|P_i - G_i|\,, \tag{7}$$

where P_i and G_i are the prediction crowd number and the corresponding ground truth of the i-th image, respectively. M is the batch size of training images.

4 Experiments

4.1 Implementation details

The transformer-encoder is similar to ViT [11], which contains 12 transformer layers, and each MSA consists of 12 SA. We utilize the fixed H and W, both of which are set as 384. We set K as 16, which means N is equal to 576. We use Adam [47] to optimize our model, in which the learning rate and weight decay are set to 1E−5 and 1E−4, respectively. The weights pre-trained on ImageNet are used to initialize the transformer-encoder. During training, the widely adopted data augmentation strategies are utilized, including random horizontal flipping and grayscaling. Due to some datasets having various resolution images, we resize all the images to the size of 1152×768. Each resized image can be regarded as six independent sub-image, and the resolution of each sub-image is 384×384. We set the batch size as 24 and use a V100 GPU for the experiments. The code is available at the website[2)].

4.2 Dataset

NWPU-Crowd [48], a large-scale and challenging dataset, consists of 5109 images, 2133375 instances annotated elaborately. To be specific, the images are randomly split into three parts, including training, validation, and testing sets, which contain 3109, 500, and 1500 images, respectively.

JHU-CROWD++ [49] 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 25791.

UCF-QNRF [50] contains 1535 images captured from unconstrained crowd scenes with about one million annotations. It has a count range of 49 to 12865, with an average count of 815.4. Specifically, the training set consists of 1201 images, and the testing set consists of 334 images.

ShanghaiTech [1] contains 1198 crowd images with 330165 annotations. The images of the dataset are divided into two parts: Part A and Part B. In particular, Part A contains 300 training images and 182 testing images, and Part B consists of 400 training images and 316 testing images.

UCF_CC_50 [51] is a small dataset for dense crowd counting, which just contains 50 images with an average of 1280 individuals per image. The images are captured in a diverse set of events, and the pedestrians count of each image ranges between 94 and 4543.

WorldExpo'10 [52] contains 1132 surveillance videos from 108 cameras. The training set consists of 3380 images captured from 103 different scenes, and the testing set contains 600 images from 5 scenes. There are 199923 annotations labeled in the whole 3980 images.

2) https://github.com/dk-liang/TransCrowd.

Table 1 Quantitative comparison (in terms of MAE and MSE) of the proposed method and some popular methods on three widely adopted benchmark datasets[a)]

Method	Year	Training label		UCF-QNRF		Part A		Part B	
		Location	Crowd number	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [1]	CVPR16	√	√	277.0	426.0	110.2	173.2	26.4	41.3
CL [50]	ECCV18	√	√	132.0	191.0	–	–	–	–
CSRNet [2]	CVPR18	√	√	–	–	68.2	115.0	10.6	16.0
L2R [15]	TPAMI19	√	√	124.0	196.0	73.6	112.0	13.7	21.4
CFF [53]	ICCV19	√	√	–	–	65.2	109.4	7.2	12.2
PGCNet [54]	ICCV19	√	√	–	–	57.0	**86.0**	8.8	13.7
TEDNet [30]	CVPR19	√	√	113.0	188.0	64.2	109.1	8.2	12.8
BL [26]	ICCV19	√	√	88.7	154.8	62.8	101.8	7.7	12.7
ASNet [27]	CVPR20	√	√	91.5	159.7	57.7	90.1	–	–
LibraNet [55]	ECCV20	√	√	88.1	143.7	**55.9**	97.1	7.3	11.3
NoisyCC [56]	NeurIPS20	√	√	85.8	150.6	61.9	99.6	7.4	11.3
DM-Count [56]	NeurIPS20	√	√	85.6	148.3	59.7	95.7	7.4	11.8
Method in [31]	MM20	√	√	84.7	147.2	58.1	91.7	6.5	**10.1**
S3 [35]	IJCAI21	√	√	**80.6**	**139.8**	57.0	96.0	**6.3**	10.6
UOT [36]	AAAI21	√	√	83.3	142.3	58.1	95.9	6.5	10.2
Method in [6]*	ECCV20	–	√	–	–	104.6	145.2	12.3	21.2
MATT [9]*	PR21	–	√	–	–	80.1	129.4	11.7	17.5
TransCrowd-Token (ours)*	–	–	√	98.9	176.1	69.0	116.5	10.6	19.7
TransCrowd-GAP (ours)*	–	–	√	**97.2**	**168.5**	**66.1**	**105.1**	**9.3**	**16.1**

a) * represents the weakly-supervised methods.

4.3 Evaluation metrics

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

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|P_i - G_i|, \quad \mathrm{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}|P_i - G_i|^2}, \tag{8}$$

where N is the number of testing images, P_i and G_i are the predicted and ground-truth counts of the i-th image, respectively.

5 Results

We conduct extensive experiments to demonstrate the effectiveness of the proposed weakly-supervised crowd counting method on five popular benchmarks. For each dataset, we divide the existing methods into fully-supervised methods (based on point-level annotations) and weakly-supervised methods (based on count-level annotations).

Compared with the weakly-supervised counting methods. Our method achieves state-of-the-art counting performance on all the conducted datasets, as listed in Tables 1–7 [53–70]. Specifically, on ShanghaiTech Part A, our TransCrowd-GAP improves 17.5% in MAE and 18.8% in MSE compared with MATT [9], improves 36.8% in MAE and 27.6% in MSE compared with [6]. On ShanghaiTech Part B, TransCrowd-GAP improves 20.5% in MAE and 8.0% in MSE compared with MATT [9], improves 24.4% in MAE and 24.1% in MSE compared with [6]. Besides, the proposed TransCrowd-Token also achieves significant improvement compared with MATT [9] and [6] in terms of MAE and MSE, and only the proposed methods report counting performance close to the fully-supervised methods. Note that MATT [9] still applies a small number of images, which contain point-level annotations for training.

Compared with the fully-supervised counting methods. Although it is unfair to compare the fully-supervised and weakly-supervised crowd counting methods, our method still achieves highly competitive performance on the five counting datasets, as shown in Tables 1–7. An impressive phenomenon is that the proposed method even surpasses some popular fully-supervised methods. For example, as shown in Table 3, our TransCrowd-GAP brings 11.0 MAE and 13.6 MSE improvement compared with CSRNet [2] on the JHU-CROWD++ (testing set) dataset. BL [26], a recent strong counting method, achieves

Table 2 Quantitative results on the JHU-CROWD++ (val set) dataset[a)]

Method	Year	Training label		Val set							
				Low		Medium		High		Overall	
		Location	Crowd number	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [1]	CVPR16	√	√	90.6	202.9	125.3	259.5	494.9	856.0	160.6	377.7
CMTL [57]	AVSS17	√	√	50.2	129.2	88.1	170.7	583.1	986.5	138.1	379.5
DSSI-Net [58]	ICCV19	√	√	50.3	85.9	82.4	164.5	436.6	814.0	116.6	317.4
CAN [59]	CVPR19	√	√	34.2	69.5	65.6	115.3	336.4	**619.7**	89.5	239.3
SANet [60]	ECCV18	√	√	13.6	26.8	50.4	78.0	397.8	749.2	82.1	272.6
CSRNet [2]	CVPR18	√	√	22.2	40.0	49.0	99.5	302.5	669.5	72.2	249.9
CG-DRCN [49]	PAMI20	√	√	17.1	44.7	40.8	**71.2**	317.4	719.8	67.9	262.1
MBTTBF [61]	ICCV19	√	√	23.3	48.5	53.2	119.9	294.5	674.5	73.8	256.8
SFCN [16]	CVPR19	√	√	11.8	19.8	**39.3**	73.4	297.3	679.4	62.9	247.5
BL [26]	ICCV19	√	√	**6.9**	**10.3**	39.7	85.2	**279.8**	620.4	**59.3**	**229.2**
TransCrowd-Token (ours)*	–	–	√	7.1	10.7	**33.3**	**54.6**	302.5	557.4	58.4	201.1
TransCrowd-GAP (ours)*	–	–	√	**6.7**	**9.5**	34.5	55.8	**285.9**	**532.8**	**56.8**	**193.6**

a) * "Low", "Medium", and "High" respectively indicate three categories based on different ranges: [0, 50], (50, 500], and >500.
* represents the weakly-supervised crowd counting methods.

Table 3 Quantitative results on the JHU-CROWD++ (testing set) dataset[a)]

Method	Year	Training label		Testing set							
				Low		Medium		High		Overall	
		Location	Crowd number	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [1]	CVPR16	√	√	97.1	192.3	121.4	191.3	618.6	1,166.7	188.9	483.4
CMTL [57]	AVSS17	√	√	58.5	136.4	81.7	144.7	635.3	1,225.3	157.8	490.4
DSSI-Net [58]	ICCV19	√	√	53.6	112.8	70.3	108.6	525.5	1,047.4	133.5	416.5
CAN [59]	CVPR19	√	√	37.6	78.8	56.4	86.2	384.2	789.0	100.1	314.0
SANet [60]	ECCV18	√	√	17.3	37.9	46.8	69.1	397.9	817.7	91.1	320.4
CSRNet [2]	CVPR18	√	√	27.1	64.9	43.9	71.2	356.2	784.4	85.9	309.2
CG-DRCN [49]	PAMI20	√	√	19.5	58.7	38.4	62.7	367.3	837.5	82.3	328.0
MBTTBF [61]	ICCV19	√	√	19.2	58.8	41.6	66.0	352.2	760.4	81.8	299.1
SFCN [16]	CVPR19	√	√	16.5	55.7	38.1	59.8	341.8	758.8	77.5	297.6
BL [26]	ICCV19	√	√	**10.1**	32.7	34.2	54.5	352.0	768.7	75.0	299.9
UOT [36]	AAAI21	√	√	11.2	**26.2**	**28.7**	**45.3**	**274.1**	**648.2**	60.5	252.7
S3 [35]	IJCAI21	√	√	–	–	–	–	–	–	**59.4**	**244.0**
TransCrowd-Token (ours)*	–	–	√	8.5	23.2	**33.3**	**71.5**	368.3	816.4	76.4	319.8
TransCrowd-GAP (ours)*	–	–	√	**7.6**	**16.7**	34.8	73.6	**354.8**	**752.8**	**74.9**	**295.6**

a) * "Low", "Medium", and "High" respectively indicate three categories based on different ranges: [0, 50], (50, 500], and >500.
* represents the weakly-supervised crowd counting methods.

75.0 in MAE and 299.9 in MSE, one of the state-of-the-art methods on the JHU-Crowd++ (testing set) dataset, while our TransCrowd-GAP improves 0.1 MAE and 4.3 MSE, respectively. Besides, from the results on UCF-QNRF, ShanghaiTech, and NWPU-Crowd datasets, we can also observe that our method achieves significant improvement compared with some popular fully-supervised methods (e.g., MCNN [1], CSRNet [2], L2R [15]). We think the reasons why the proposed method outperforms some fully supervised methods in the NWPU-Crowd and JHU-CROWD++ may be two-fold. First, the transformer is beneficial to capture the long-range dependence, and these two datasets contain many large-scale persons. The proposed TransCrowd can effectively learn the global crowd semantic feature representation. However, some state-of-the-art methods (e.g., BL [26]) utilize a fixed Gaussian kernel for these datasets, and the fixed Gaussian kernel cannot effectively cover the large scale variations. Second, Dosovitskiy et al. [11] proved that the CNNs outperform transformers on small datasets (despite regularization optimization), but with the larger datasets, the transformer overtakes. For instance, the NWPU-Crowd is a large dataset containing 5190 images, which may help the transformer better fit the dataset. These impressive results further demonstrate the effectiveness of the proposed method and indicate point-level annotations are not entirely necessary for the counting task.

Table 4 Comparison of the counting performance on the NWPU-Crowd[a)]

Method	Year	Training label		Val set		Testing set			
				Overall		Overall		Scene level (only MAE)	
		Location	Crowd number	MAE	MSE	MAE	MSE	Average	*S*0–*S*4
C3F-VGG [62]	Tech19	√	√	105.79	504.39	127.0	439.6	666.9	140.9/26.5/58.0/307.1/2801.8
CSRNet [2]	CVPR18	√	√	104.89	433.48	121.3	387.8	522.7	176.0/35.8/59.8/285.8/2055.8
PCC-Net-VGG [33]	CVPR19	√	√	100.77	573.19	112.3	457.0	777.6	103.9/13.7/42.0/259.5/3469.1
CAN [59]	CVPR19	√	√	93.58	489.90	106.3	386.5	612.2	82.6/14.7/46.6/269.7/2647.0
SFCN† [16]	CVPR19	√	√	95.46	608.32	105.7	424.1	712.7	54.2/14.8/44.4/249.6/3200.5
BL [26]	ICCV19	√	√	93.64	470.38	105.4	454.2	750.5	66.5/8.7/41.2/249.9/3386.4
KDMG [63]	PAMI20	√	√	–	–	100.5	415.5	632.7	77.3/10.3/38.5/259.4/2777.9
NoisyCC [56]	NeurIPS20	√	√	–	–	96.9	534.2	608.1	218.7/10.7/35.2/203.2/2572.8
DM-Count [64]	NeurIPS20	√	√	**70.5**	**357.6**	88.4	388.6	**498.0**	146.6/7.6/31.2/228.7/2075.8
S3 [35]	IJCAI21	√	√	–	–	87.8	387.5	566.5	80.7/7.9/36.3/212.0/2495.4
UOT [36]	AAAI21	√	√	–	–	**83.5**	**346.9**	–	–
TransCrowd-Token (ours)*	–	–	√	**88.2**	446.9	119.6	463.9	**736.0**	88.0/12.7/47.2/311.2/3216.1
TransCrowd-GAP (ours)*	–	–	√	88.4	**400.5**	**117.7**	**451.0**	737.8	69.3/12.8/46.0/309.0/3252.2

a) * *S*0–*S*4 respectively indicate five categories according to the different number ranges: 0, $(0, 100]$, $(100, 500]$, $(500, 5000]$, >5000. * represents the weakly-supervised crowd counting methods.

Table 5 The performance comparison on the UCF_CC_50 dataset[a)]

Method	Training label		MAE	MSE
	Location	Crowd number		
MCNN [65]	√	√	377.6	509.1
CSRNet [2]	√	√	266.1	397.5
ADCrowdNet [37]	√	√	**257.1**	**363.5**
MATT [9]*	–	√	355.0	550.2
TransCrowd-Token (ours)*	–	√	288.9	407.6
TransCrowd-GAP (ours)*	–	√	**272.2**	**395.3**

a) * represents the weakly-supervised crowd counting methods.

Table 6 Comparison results of different methods on 5 scenes in the WorldExpo'10 dataset[a)]

Method	Training label		MAE					
	Location	Crowd number	S1	S2	S3	S4	S5	Average
MCNN [1]	√	√	3.4	20.6	12.9	13.0	8.1	11.6
CP-CNN [29]	√	√	2.9	14.7	10.5	10.4	5.8	8.8
Liu et al. [66]	√	√	**2.0**	13.1	8.9	17.4	4.8	9.2
IC-CNN [67]	√	√	17.0	12.3	9.2	**8.1**	4.7	10.3
CSRNet [2]	√	√	2.9	11.5	**8.6**	16.6	3.4	8.6
SANet [60]	√	√	2.6	13.2	9.0	13.3	**3.0**	8.2
LSC-CNN [68]	√	√	2.9	**11.3**	9.4	12.3	4.3	**8.0**
MATT [9]*	–	√	3.8	**13.1**	10.4	15.9	5.3	9.7
TransCrowd-Token (ours)*	–	√	2.3	14.2	9.9	14.0	**4.3**	8.9
TransCrowd-GAP (ours)*	–	√	**2.1**	13.3	**8.9**	**13.8**	4.4	**8.5**

a) * represents the weakly-supervised crowd counting methods. S1, S2, S3, S4, and S5 indicate different scenes.

Table 7 The performance comparison on the Trancos dataset[a)]

Method	Training label		MAE	MSE
	Location	Crowd number		
FCN-HA [65]	√	√	4.21	–
CSRNet [2]	√	√	3.56	–
ADCrowdNet [37]	√	√	**2.44**	–
TransCrowd-Token (ours)*	–	√	3.28	4.80
TransCrowd-GAP (ours)*	–	√	**3.23**	**4.66**

a) * represents the weakly-supervised crowd counting methods.

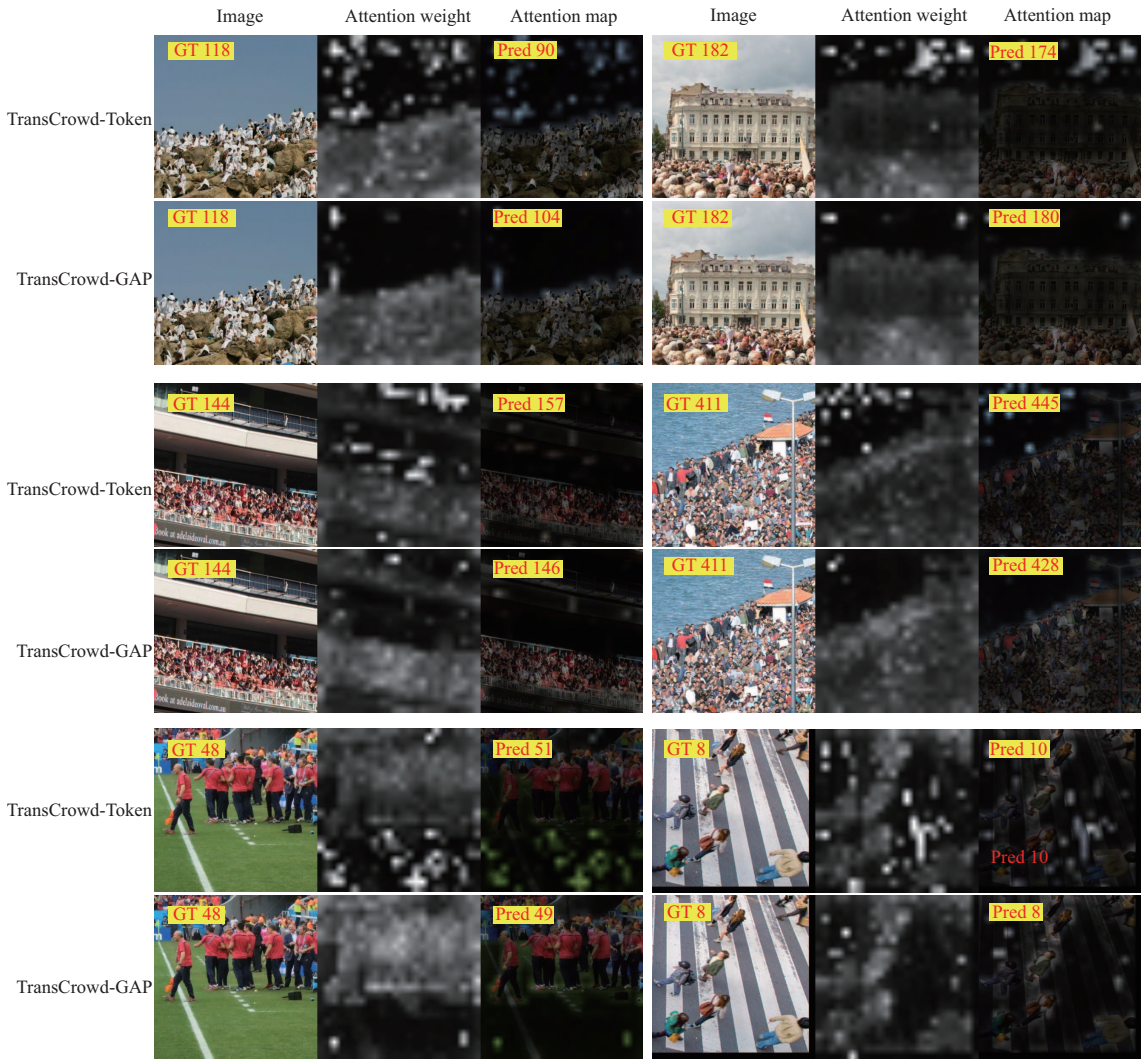

Figure 4 (Color online) Examples of attention maps from TransCrowd-Token and TransCrowd-GAP. TransCrowd-GAP generates more reasonable attention weights compared with TransCrowd-Token.

6 Analysis

6.1 The influence of regression heads

We introduce two different inputs for the regression head. Specifically, TransCrowd-Token utilizes an extra learnable regression token to perform counting, similar to the class token in Bert [42] and ViT [11]. TransCrowd-GAP utilizes global average pooling to obtain the pooled visual tokens for count prediction. The results of TransCrowd-Token and TransCrowd-GAP are listed in Tables 1–7. We find that the results of TransCrowd-GAP are better than TransCrowd-Token in all conducted datasets. For example, TransCrowd-GAP outperforms TransCrowd-Token by 2.8 MAE and 11.4 MSE on the ShanghaiTech Part A dataset, a significant improvement. TransCrowd-GAP also has steady improvement on ShanghaiTech Part B, a sparse crowd dataset. Based on the superior performance, we hope the researchers can design a more reasonable regression head based on the transformer-encoder in the future.

6.2 Visualizations

To further investigate the proposed TransCrowd, we provide qualitative comparison results in Figure 4 to understand what the transformer attends to. We observe that both TransCrowd-Token and TransCrowd-GAP can successfully focus on the crowd region, which demonstrates the effectiveness of both methods. Moreover, the TransCrowd-GAP generates a more reasonable attention map compared with the

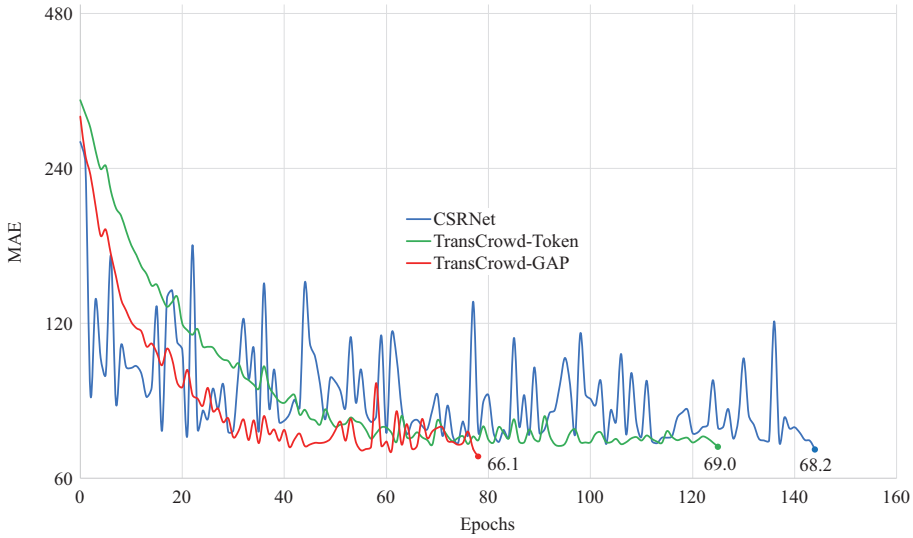

Figure 5 (Color online) Convergence curves of CSRNet, TransCrowd-Token, and TransCrowd-GAP on ShanghaiTech Part A dataset. The proposed TransCrowd-GAP achieves the best counting performance and is fast-converging.

Table 8 Comparison with BL [26] and CSRNet [2] using the same input image resolution on a Titan XP

Method	Resolution	Parameter	Backbone	FPS
CSRNet [2]	384 × 384	16.2 M	VGG16	21.67
BL [26]	384 × 384	21.6 M	VGG19	45.66
TransCrowd-Token	384 × 384	86.8 M	Transformer	46.41
TransCrowd-GAP	384 × 384	90.4 M	Transformer	46.73

TransCrowd-Token. Specifically, the TransCrowd-Token may pay more attention to the background, leading to amplifying the counting error. This observation explains why the result of TransCrowd-GAP is better than TransCrowd-Token.

6.3 Convergence curves

We further compare the convergence curves between the popular fully-supervised method (CSRNet [2]) and the proposed TransCrowd. Detailed convergence curves are shown in Figure 5. Based on the convergence curves, we can observe the following phenomena: (1) Compared with CSRNet, TransCrowd-GAP achieves better performance with 1.9× fewer training epochs. (2) Using global average pooled visual tokens can converge faster and achieve a better count accuracy than using the extra regression token. (3) Both TransCrowd-Token and TransCrowd-GAP present a smooth curve and fast converging, while the curve of CSRNet is oscillating. These observations show the potential value of the transformer in the counting task.

6.4 Comparison of run-time

As shown in Table 8, we compare with two popular fully supervised counting methods, including BL [26] and CSRNet [2]. The experiment is conducted on a Titan XP GPU. Even though both the proposed TransCrowd-Token and TransCrowd-GAP contain more parameters than other methods, they still achieve outstanding run-time. This is because the fully supervised methods need to maintain high-resolution features to generate high-quality density maps (e.g., 1/8 size of the input in CSRNet [2] and 1/16 size of the input in BL [26]). Additionally, we can observe that the frames per second (FPS) of VGG19-based BL [26] outperforms the VGG16-based CSRNet [2], 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.

Table 9 The fine-tuning CSRNet's and TransCrowd-GAP's results on ShanghaiTech Part A dataset by using three different pre-trained strategies

Method	Year	Training label		None		Pre-ImgNet		Pre-GCC	
		Location	Crowd number	MAE	MSE	MAE	MSE	MAE	MSE
CSRNet [2]	CVPR18	√	√	**120.0**	**179.4**	68.2	115.0	67.4	112.3
TransCrowd-Token (ours)*	–	–	√	142.0	212.5	69.0	116.5	67.2	111.9
TransCrowd-GAP (ours)*	–	–	√	139.9	231.0	**66.1**	**105.1**	**63.8**	**102.3**

Table 10 Experimental results on the transferability of different methods under cross-dataset evaluation

Method	Year	Training label		Part B→Part A		Part A→Part B		QNRF→Part A		QNRF→Part B	
		Location	Crowd number	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
MCNN [1]	CVPR16	√	√	221.4	357.8	85.2	142.3	–	–	–	–
D-ConvNet [69]	ECCV18	√	√	**140.4**	**226.1**	49.1	99.2	–	–	–	–
RRSP [70]	CVPR19	√	√	–	–	**40.0**	**68.5**	–	–	–	–
BL [26]	ICCV19	√	√	–	–	–	–	**69.8**	**123.8**	**15.3**	**26.5**
TransCrowd-GAP (ours)*	–	–	√	**141.3**	**258.9**	**18.9**	**31.1**	**78.7**	**122.5**	**13.5**	**21.9**

6.5 Comparison of different pre-trained strategies

In this subsection, we study the impact of the pre-trained model in TransCrowd. We choose the popular CNN-based method CSRNet [2] as a comparison, and the results are listed in Table 9. Specifically, there are three strategies. (1) None: The models are directly trained on ShanghaiTech Part A. (2) Pre-ImgNet: The models are pre-trained on the ImageNet and fine-tuned on ShanghaiTech Part A. (3) Pre-GCC: The models are pre-trained on GCC [16], a synthetic dataset, and are fine-tuned on ShangahiTech Part A dataset.

From Table 9, there are some interesting findings. (1) Without any pre-trained dataset, the CNN-based method outperforms the transformer-based method. (2) Using the extra pre-trained data can effectively prompt the performance, and the proposed TransCrowd-GAP achieves better counting performance than CSRNet. (3) Besides, when the model is pre-trained on the GCC dataset, the proposed method can even outperform several recent fully-supervised methods (e.g., CFF [53], TEDNet [30]). Note that the GCC dataset is a synthetic crowd dataset, without any annotation cost, which means the TransCrowd-GAP can achieve similar counting performance to the fully-supervised methods by using small count-level labeled real-data and extensive free synthetic data, promoting the practical applications. It is noteworthy that the proposed method only uses count-level annotations of the GCC dataset, different from the previous fully-supervised work.

6.6 Cross-dataset evaluation

Finally, we conduct cross-dataset experiments on the UCF-QNRF, ShanghaiTech Part A and Part B datasets to explore the transferability of the proposed TransCrowd-GAP. In the cross-dataset evaluation, models are trained on the source dataset and tested on the target dataset without further fine-tuning. Quantitative results are shown in Table 10. Although our method is a weakly-supervised paradigm, we still achieve highly competitive performance, which shows remarkable transferability.

7 Conclusion

In this study, we present an alternative perspective for weakly-supervised crowd counting in images by introducing a sequence-to-count prediction framework based on transformer-encoder, named TransCrowd. To the best of our knowledge, we are the first to solve the counting problem based on the transformer. We analyze and show that the attention mechanism is very promising to capture the semantic crowd information. Extensive experiments on five challenging datasets demonstrate that TransCrowd achieves superior counting performance compared with the state-of-the-art weakly-supervised methods and achieves competitive performance compared with some popular fully-supervised methods. In the future, we plan to make fully-supervised counting using transformer architectures and extend it to video-based counting task.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2018YFB1004600).

References

1 Zhang Y, Zhou D, Chen S, et al. Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2016
2 Li Y, Zhang X, Chen D. CSRNet: dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2018
3 Xu C, Qiu K, Fu J, et al. Learn to scale: generating multipolar normalized density map for crowd counting. In: Proceedings of IEEE International Conference on Computer Vision, 2019
4 Liu Z, He Z, Wang L, et al. VisDrone-CC2021: the vision meets drone crowd counting challenge results. In: Proceedings of IEEE International Conference on Computer Vision, 2021. 2830–2838
5 Bai S, He Z, Qiao Y, et al. Adaptive dilated network with self-correction supervision for counting. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2020
6 Yang Y, Li G, Wu Z, et al. Weakly-supervised crowd counting learns from sorting rather than locations. In: Proceedings of European Conference on Computer Vision, 2020
7 Guo B, Wang Z, Yu Z, et al. Mobile crowd sensing and computing. ACM Comput Surv, 2015, 48: 1–31
8 Sheng X, Tang J, Xiao X J, et al. Leveraging GPS-less sensing scheduling for green mobile crowd sensing. IEEE Internet Things J, 2014, 1: 328–336
9 Lei Y, Liu Y, Zhang P, et al. Towards using count-level weak supervision for crowd counting. Pattern Recogn, 2021, 109: 107616
10 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of Advances in Neural Information Processing Systems, 2017
11 Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16 × 16 words: transformers for image recognition at scale. 2021. ArXiv:2010.11929
12 Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers. In: Proceedings of European Conference on Computer Vision, 2020
13 Zhu X, Su W, Lu L, et al. Deformable DETR: deformable transformers for end-to-end object detection. In: Proceedings of International Conference on Learning Representations, 2020
14 Zheng S, Lu J, Zhao H, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2021. 6881–6890
15 Liu X, Weijer J, Bagdanov A D. Exploiting unlabeled data in CNNs by self-supervised learning to rank. IEEE Trans Pattern Anal Mach Intell, 2019, 41: 1862–1878
16 Wang Q, Gao J, Lin W, et al. Learning from synthetic data for crowd counting in the wild. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2019
17 Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. In: Proceedings of Advances in Neural Information Processing Systems, 2015
18 Liu W, Anguelov D, Erhan D, et al. SSD: single shot multibox detector. In: Proceedings of European Conference on Computer Vision, 2016
19 Abousamra S, Hoai M, Samaras D, et al. Localization in the crowd with topological constraints. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021
20 Liu Y, Shi M, Zhao Q, et al. Point in, box out: beyond counting persons in crowds. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2019
21 Liang D, Xu W, Zhu Y, et al. Focal inverse distance transform maps for crowd localization and counting in dense crowd. 2021. ArXiv:2102.07925
22 Xu C, Liang D, Xu Y, et al. AutoScale: learning to scale for crowd counting. Int J Comput Vis, 2022, 130: 405–434
23 Chen Y, Liang D, Bai X, et al. Cell localization and counting using direction field map. IEEE J Biomed Health Inform, 2022, 26: 359–368
24 Zhang A, Yue L, Shen J, et al. Attentional neural fields for crowd counting. In: Proceedings of IEEE International Conference on Computer Vision, 2019
25 Du D, Wen L, Zhu P, et al. VisDrone-CC2020: the vision meets drone crowd counting challenge results. In: Proceedings of European Conference on Computer Vision, 2020. 675–691
26 Ma Z, Wei X, Hong X, et al. Bayesian loss for crowd count estimation with point supervision. In: Proceedings of IEEE International Conference on Computer Vision, 2019
27 Jiang X, Zhang L, Xu M, et al. Attention scaling for crowd counting. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2020
28 Xu W, Liang D, Zheng Y, et al. Dilated-scale-aware category-attention ConvNet for multi-class object counting. IEEE Signal Process Lett, 2021, 28: 1570–1574
29 Sindagi V A, Patel V M. Generating high-quality crowd density maps using contextual pyramid CNNs. In: Proceedings of IEEE International Conference on Computer Vision, 2017
30 Jiang X, Xiao Z, Zhang B, et al. Crowd counting and density estimation by trellis encoder-decoder networks. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2019
31 Ma Z, Wei X, Hong X, et al. Learning scales from points: a scale-aware probabilistic model for crowd counting. In: Proceedings of ACM Multimedia, 2020. 220–228
32 Shi M, Yang Z, Xu C, et al. Revisiting perspective information for efficient crowd counting. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2019
33 Gao J, Wang Q, Li X. PCC Net: perspective crowd counting via spatial convolutional network. IEEE Trans Circuits Syst Video Technol, 2020, 30: 3486–3498
34 Yang Y, Li G, Wu Z, et al. Reverse perspective network for perspective-aware object counting. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2020
35 Lin H, Hong X, Ma Z, et al. Direct measure matching for crowd counting. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence, 2021

36 Ma Z, Wei X, Hong X, et al. Learning to count via unbalanced optimal transport. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2021. 2319–2327
37 Liu N, Long Y, Zou C, et al. ADCrowdNet: an attention-injective deformable convolutional network for crowd understanding. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2019
38 Chan A B, Liang Z S J, Vasconcelos N. Privacy preserving crowd monitoring: counting people without people models or tracking. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2008
39 von Borstel M, Kandemir M, Schmidt P, et al. Gaussian process density counting from weak supervision. In: Proceedings of European Conference on Computer Vision. Springer, 2016. 365–380
40 Shang C, Ai H, Bai B. End-to-end crowd counting via joint learning local and global count. In: Proceedings of IEEE International Conference on Image Processing, 2016
41 Wang C, Zhang H, Yang L, et al. Deep people counting in extremely dense crowds. In: Proceedings of ACM Multimedia, 2015
42 Devlin J, Chang M-W, Toutanova L K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2019. 4171–4186
43 Liu Y, Ott M, Goyal N, et al. RoBERTa: a robustly optimized bert pretraining approach. 2019. ArXiv:1907.11692
44 Wang N, Zhou W, Wang J, et al. Transformer meets tracker: exploiting temporal context for robust visual tracking. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2021. 1571–1580
45 Chen H, Wang Y, Guo T, et al. Pre-trained image processing transformer. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2021. 12299–12310
46 Hendrycks D, Gimpel K. Bridging nonlinearities and stochastic regularizers with Gaussian error linear units. 2016. ArXiv:1606.08415
47 Kingma D, Ba J. Adam: a method for stochastic optimization. In: Proceedings of International Conference on Learning Representations, 2015
48 Wang Q, Gao J, Lin W, et al. NWPU-Crowd: a large-scale benchmark for crowd counting and localization. IEEE Trans Pattern Anal Mach Intell, 2021, 43: 2141–2149
49 Sindagi V A, Yasarla R, Patel V M. JHU-CROWD++: large-scale crowd counting dataset and a benchmark method. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 2594–2609
50 Idrees H, Tayyab M, Athrey K, et al. Composition loss for counting, density map estimation and localization in dense crowds. In: Proceedings of European Conference on Computer Vision, 2018
51 Idrees H, Saleemi I, Seibert C, et al. Multi-source multi-scale counting in extremely dense crowd images. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2013
52 Zhang C, Li H, Wang X, et al. Cross-scene crowd counting via deep convolutional neural networks. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2015. 833–841
53 Shi Z, Mettes P, Snoek C G. Counting with focus for free. In: Proceedings of IEEE International Conference on Computer Vision, 2019. 4200–4209
54 Yan Z, Yuan Y, Zuo W, et al. Perspective-guided convolution networks for crowd counting. In: Proceedings of IEEE International Conference on Computer Vision, 2019
55 Liu L, Lu H, Zou H, et al. Weighing counts: sequential crowd counting by reinforcement learning. In: Proceedings of European Conference on Computer Vision. Springer, 2020. 164–181
56 Wan J, Chan A. Modeling noisy annotations for crowd counting. In: Proceedings of Advances in Neural Information Processing Systems, 2020
57 Sindagi V A, Patel V M. CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In: Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance, 2017
58 Liu L, Qiu Z, Li G, et al. Crowd counting with deep structured scale integration network. In: Proceedings of IEEE International Conference on Computer Vision, 2019
59 Liu W, Salzmann M, Fua P. Context-aware crowd counting. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2019
60 Cao X, Wang Z, Zhao Y, et al. Scale aggregation network for accurate and efficient crowd counting. In: Proceedings of European Conference on Computer Vision, 2018
61 Sindagi V A, Patel V M. Multi-level bottom-top and top-bottom feature fusion for crowd counting. In: Proceedings of IEEE International Conference on Computer Vision, 2019
62 Gao J, Lin W, Zhao B, et al. C^3 framework: an open-source PyTorch code for crowd counting. 2019. ArXiv:1907.02724
63 Wan J, Wang Q, Chan A B. Kernel-based density map generation for dense object counting. IEEE Trans Pattern Anal Mach Intell, 2022, 44: 1357–1370
64 Wang B, Liu H, Samaras D, et al. Distribution matching for crowd counting. In: Proceedings of Advances in Neural Information Processing Systems, 2020
65 Zhang S, Wu G, Costeira J P, et al. FCN-rLSTM: deep spatio-temporal neural networks for vehicle counting in city cameras. In: Proceedings of IEEE International Conference on Computer Vision, 2017
66 Liu J, Gao C, Meng D, et al. DecideNet: counting varying density crowds through attention guided detection and density estimation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2018
67 Ranjan V, Le H, Hoai M. Iterative crowd counting. In: Proceedings of European Conference on Computer Vision, 2018
68 Sam D B, Peri S V, Sundararaman M N, et al. Locate, size and count: accurately resolving people in dense crowds via detection. IEEE Trans Pattern Anal Mach Intell, 2021, 43: 2739–2751
69 Shi Z, Zhang L, Liu Y, et al. Crowd counting with deep negative correlation learning. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2018
70 Wan J, Luo W, Wu B, et al. Residual regression with semantic prior for crowd counting. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, 2019