# TransCrowd: Weakly-Supervised Crowd Counting with Transformer

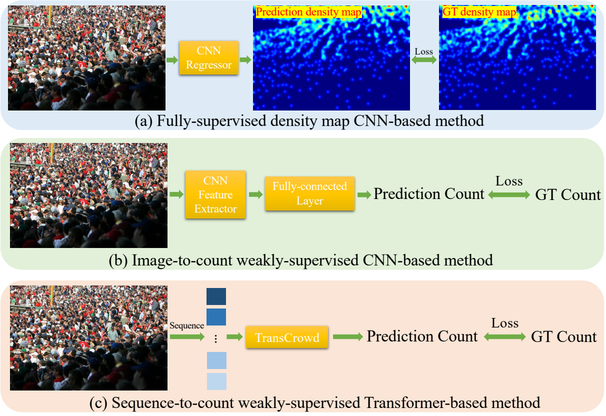
Dingkang Liang[1], Xiwu Chen[1], Wei Xu[2], Yu Zhou[1], Xiang Bai[1]
[1]Huazhong University of Science and Technology
[2]Beijing University of Posts and Telecommunications

## 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 can not achieve satisfactory performance, limited applications in the real-word. The Transformer is a popular sequence-to-sequence prediction model in 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 Transformer. 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. Code is available at `https://github.com/dk-liang/TransCrowd`.*

## 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 [70, 23, 27, 62, 2] 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 decoder: the encoder extracts the high-level feature information, and the decoder is designed for pixel-level regression based on the extracted feature.



(a) Fully-supervised density map CNN-based method

(b) Image-to-count weakly-supervised CNN-based method

(c) Sequence-to-count weakly-supervised Transformer-based method

Figure 1: (a) Traditional fully-supervised CNN-based method. All the training images are labeled with point-level annotations. (b) Weakly-supervised CNN-based method from 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.

However, these density-map regression-based methods [70, 23, 27, 62, 2] 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 [67, 15, 40, 21] discover that we can collect a new crowd dataset by using a more economical strategy, such as mobile crowd-sensing [15] technology or GPS-less [40] energy-efficient sensing scheduling. For a given crowd scene with different viewpoints and the total count keeps the same (such as auditoria, classroom), if we know the total count of one viewpoint, then the total count of other viewpoints is 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 evaluating the counting performance, meaning the point 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 works [21, 67], we call the methods which rely on the point-level annotations are fully-supervised paradigm, and the methods which only rely on count-level are weakly-supervised paradigm. 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 $L_2$ loss to measure the difference between the prediction and the ground truth, as shown in Fig. 1(a). The existing weakly-supervised methods usually regress the total count of crowd image directly, which is from the image-to-count perspective, as shown in Fig. 1(b).

Recently, Transformer [49], a popular language model proposed by Google, has been explored in many vision tasks, such as classification [11], detection [4, 72], and segmentation [71]. Unlike the CNN, which utilizes a limited receptive field, the Transformer [49] 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 Fig. 1(c).

Only a few methods are proposed with the considerations of reducing the annotations burden (e.g., semi, weakly-supervised). L2R [31] facilitates the counting task by ranking the image patch, applying the location labels. Wang *et al.* [57] introduce 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 works for our method is [67], which proposing a soft-label network to facilitate the counting task, directly regressing the crowd number without the supervision of location labels. However, [67] 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 higher count accuracy, and present fast-converging compared with TransCrowd-Token.

In summary, this work contributes to the following:

1. TransCrowd is the first 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.

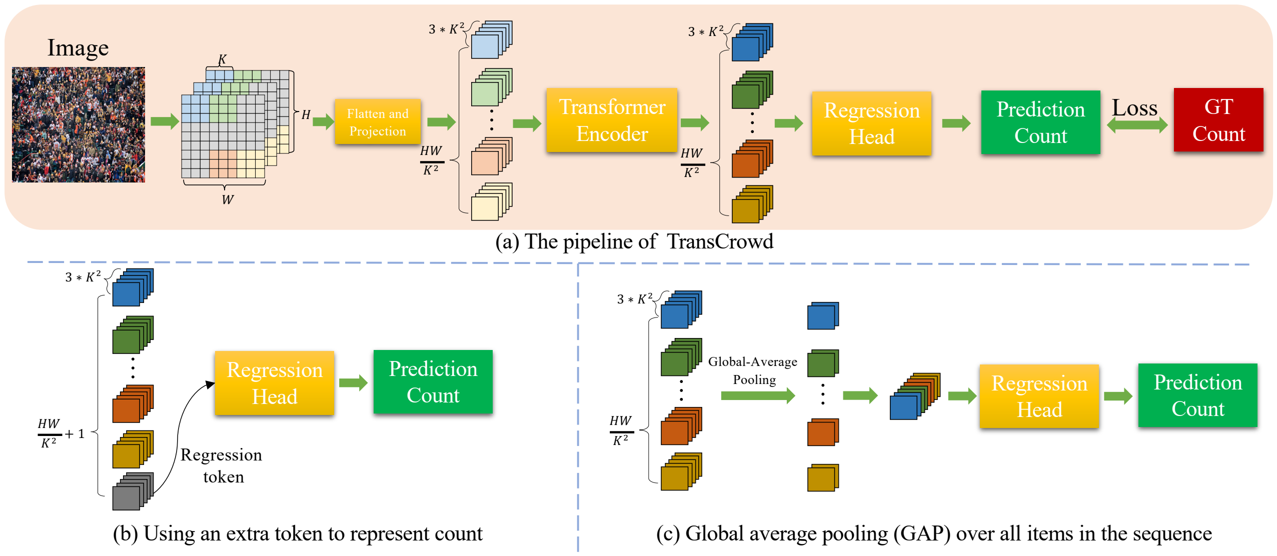
2. We provide two different types of TransCrowd, named TransCrowd-Token and TransCrowd-GAP, respectively. We observe that the TransCrowd-GAP can generate a more reasonable attention weight while reports faster converging and higher counting performance than TransCrowd-Token.

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 Works

### 2.1. CNN-based crowd counting.

The CNN-based crowd counting methods can be categorized into localization-based methods and density map regression-based methods. The localization-based methods [37, 29] usually learn to predict bounding boxes for each human, rely on box-level annotations. Recently, some methods [1, 33, 38, 24, 61, 62] 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 [70, 23, 69, 12, 61, 34, 19, 63], whose integral of the density map gives the total count of a crowd image. Due to the commonly heavy occlusion that existed in crowd images, multi-scale architecture is developed. Specifically, MCNN [70] utilizes multi-size filters to extract different scale feature information. Sam *et al.* [46] capture the multi-scale information by the proposed contextual pyramid CNN. TEDNet [18] assembles multiple



(a) The pipeline of TransCrowd

(b) Using an extra token to represent count

(c) Global average pooling (GAP) over all items in the sequence

Figure 2: (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 [10] and ViT [11]. (c) A global average pooling is adopted to pool the output visual tokens of the Transformer-encoder.

encoding-decoding paths hierarchically to generate a high-quality density map for accurate crowd counting. Additionally, using the perspective information to diminish the scale variations is effective [64, 41, 14, 66]. PACNN [41] 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.* [66] propose a reverse perspective network to estimate the perspective factor of the input image and then warp the image.

The Attention-based mechanism is another useful technique adopted by many methods [27, 19, 69]. ADCrowdNet [27] generates an attention map for the crowd images via a network called Attention Map Generate (AMG). Jiang *et al.* [19] propose a density attention network to generate attention masks concerning regions of different density levels. Zhang *et al.* [69] propose a Relation Attention Network (RANet) that utilizes local self-attention and global self-attention to capture long-range dependencies.

## 2.2. Weakly-supervised crowd counting.

Only a few methods focus on counting with a lack of labeled data. L2R [31] proposes a learning-to-rank framework based on an extra collected crowd dataset. Wang *et al.* [57] introduce 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.

MATT [21] learns a model from a small amount of point-level annotations (fully-supervised) and a large amount of count-level annotations (weakly-supervised). Similarly, Sam *et al.* [39] proposed an almost unsupervised counting method. Specifically, most parameters of the model are trained without any labeled data, and only a few parameters are trained with point-level annotated data. The method in [50] proposes a weakly-supervised solution based on the Gaussian process for crowd density estimation. Yang *et al.* [67] directly regress 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 real-world. Different from the previous works, 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 [49], dominating the natural language modeling [10, 68, 32, 36], utilize the self-attention mechanism to capture the global dependencies between input and output. Recently, many works [4, 11, 71, 55, 9, 7, 5, 72, 28, 59] attempt to apply the Transformer into the vision task. Specifically, DETR [4] firstly utilizes a CNN backbone to extract the visual features, followed by the Transformer blocks for the box regression and classification. Based on the DETR [4], Deformable-DETR [72] further introduces the deformable convolution [8] to mitigates the slow convergence and high complexity issues of DETR [4]. ViT [11] is the first which directly applies Transformer-encoder [49] to sequences of images patch to realize classification task. SETR [71] regards semantic segmentation from a sequence-to-sequence perspective with Transformers. IPT [5] develops a pre-trained model for image processing (low-level task) using the transformer architecture. Besides, Transformer is also adopted to cope with other vision tasks, such as multiple-object tracking [35, 65, 22], image generation [58], medical image segmentation [48, 6].

To the best of our knowledge, we are the first to explore the Transformer [49] 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 Fig. 2(a). Specifically, the input image is first transformed into fixed-size patches and then flatten to a sequence of vectors. The sequence is feed 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 $1D$ sequences. Specifically, given an RGB image $I \in \mathbb{R}^{H\times W\times 3}$ where $H$, $W$, 3 indicate the spatial height, width and channel number, we divide the $I$ into a grid of $\frac{H}{K} \times \frac{W}{K}$ patches, and the size of each patch is $K \times K \times 3$. Then we flatten the grid into a sequence $x \in \mathbb{R}^{N\times D}$, where $N = \frac{HW}{K^2}$, and $D = K \times K \times 3$.

### 3.2. Patch Embedding

Next, we need to map the $x$ into a latent $D$-dimensional embedding feature by a learnable projection $f : x_i \rightarrow e_i \in \mathbb{R}^D$. To encode the patch spacial information, a specific position embedding ($p$) is adopted to maintain the position information, as below:

$$Z_0 = [e_1 + p_1, e_2 + p_2, ..., e_N + p_N], \quad (1)$$

where $Z_0 \in \mathbb{R}^{N\times D}$ is the input of Transformer-encoder.

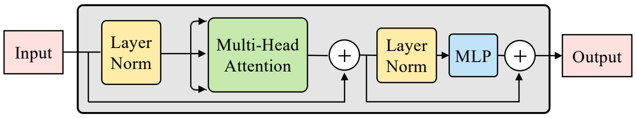


Figure 3: A standard Transformer layer consists of Multi-Head Attention and *MLP* blocks. Meanwhile, the layer normalization (*LN*) and residual connections are employed.

### 3.3. Transformer-encoder

We only adopt the Transformer encoder [49], without the decoder, same as 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 Fig. 3, and the output can be written as follows:

$$Z'_l = MSA(LN(Z_{l-1})) + Z_{l-1}, \quad (2)$$

$$Z_l = MLP(LN(Z'_l)) + Z'_l, \quad (3)$$

where $Z_l$ is the output of layer $l$. Here, the $MLP$ contains two linear layers with a GELU [16] activation function. In particular, the first linear layer of $MLP$ expands the feature embedding' 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: $MSA(Z_{l-1}) = [SA_1(Z_{l-1}); SA_2(Z_{l-1}); \cdots ; SA_m(Z_{l-1})]W_O$, 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 $Z^{l-1}$:

$$Q = Z_{l-1}W_Q, \quad K = Z_{l-1}W_K, \quad V = Z_{l-1}W_V, \quad (4)$$

$$SA(Z_l) = softmax(\frac{QK^T}{\sqrt{D}})V \quad (5)$$

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 [10] and ViT [11], we prepend a learnable embedding named regression token to the input sequence $Z_0$, as shown in Fig. 2(b). 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 Fig. 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 Sec. 6.

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

$$L_1 = \frac{1}{M}\sum_{i=1}^{M} |P_i - G_i|, \tag{6}$$

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

# 4. Experiments

## 4.1. Implementation details

The Transformer-encoder is the same as 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 [20] 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 into 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.

## 4.2. Dataset

**NWPU-Crowd** [56], a large-scale and challenging dataset, consists of 5,109 images, 2,133,375 instances annotated elaborately. To be specific, the images are randomly split into three parts, including training, validation, and testing sets, which contain 3,109, 500, and 1,500 images, respectively.

**JHU-CROWD++** [44] contains 2,722 training images, 500 validation images, and 1,600 testing images, collected from diverse scenarios. The total number of people in each image ranges from 0 to 25,791.

**UCF-QNRF** [17] contains 1,535 images captured from unconstrained crowd scenes with about one million annotations. It has a count range of 49 to 12,865, with an average count of 815.4. Specifically, the training set consists of 1,201 images, and the testing set consists of 334 images.

**ShanghaiTech** [70] contains 1,198 crowd images with 330,165 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.

## 4.3. Evaluation Metrics

We choose Mean Absolute Error (MAE) and Mean Squared Error (MSE) to evaluate the counting performance:

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

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

where $N$ is the number of testing images, $P_i$ and $G_i$ are the predicted and ground truth count 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 crowd counting methods.** Our method achieves state-of-the-art counting performance on all the conducted datasets, as listed in Tab. 1, Tab. 2, Tab. 3, and Tab. 4. Specifically, on ShanghaiTech part A, our TransCrowd-GAP improves 17.5% in MAE and 18.8% in MSE compared with MATT [21], improves 36.8% in MAE and 27.6% in MSE compared with [67]. On ShanghaiTech part B, TransCrod-GAP improves 20.5% in MAE and 8.0% in MSE compared with MATT [21], improves 24.4% in MAE and 24.1% in MSE compared with [67]. Besides, the proposed TransCrowd-Token also achieves significant improvement compared with MATT [21] and [67] in terms of MAE and MSE, and only the proposed methods report counting performance close to the fully-supervised methods. Note that MATT [21] still applies a small number of images, which contain point-level annotations for training.

**Compared with the fully-supervised crowd counting methods.** Although it is unfair to compare the fully-

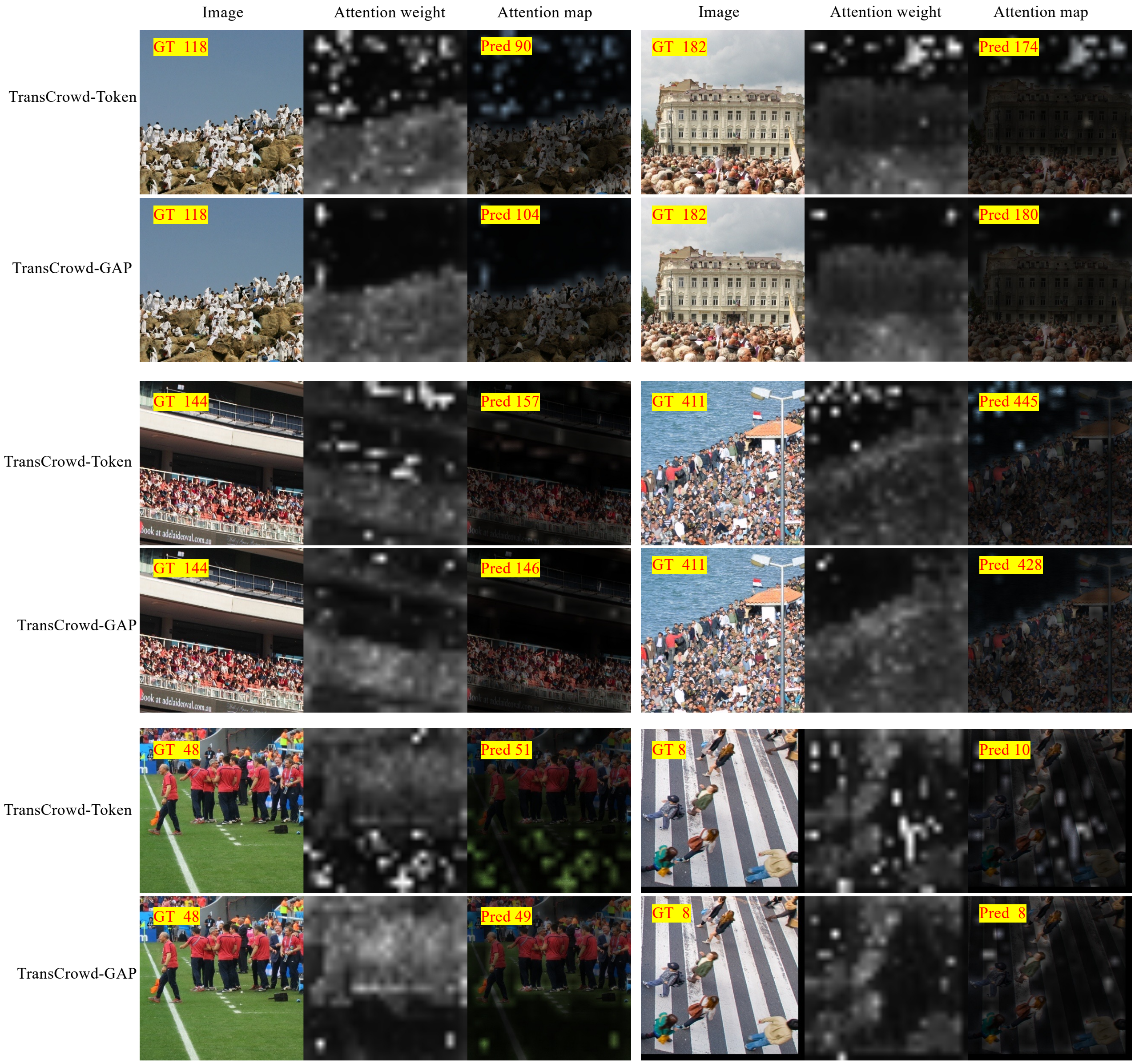


Figure 4: Examples of attention maps from TransCrowd-Token and TransCrowd-GAP. TransCrowd-GAP generates more reasonable attention weights compared with TransCrowd-Token.

supervised and weakly-supervised crowd counting methods, our method still achieves highly competitive performance on the five counting datasets, as shown in Tab. 1, Tab. 2, Tab. 3, and Tab. 4. An impressive phenomenon is that the proposed method even surpasses some popular fully-supervised methods. For example, as shown in Tab. 3, our TransCrowd-GAP brings 11.0 MAE and 13.6 MSE improvement compared with CSRNet [23] on the JHU-Crowd++ (testing set) dataset. BL [34], a recent strong counting method, achieves 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 achieve significant improvement compared to some popular fully-supervised methods (e.g., MCNN [70], CSRNet [23], L2R [31]). These impressive results further demonstrate the effectiveness of the proposed method and indicate point-level annotations are not entirely necessary for the counting task.

| Method | Year | Training label | | UCF-QNRF | | Part A | | Part B | |
|---|---|---|---|---|---|---|---|---|---|
| | | Location | Crowd number | MAE | MSE | MAE | MSE | MAE | MSE |
| MCNN [70] | CVPR16 | √ | √ | 277.0 | 426.0 | 110.2 | 173.2 | 26.4 | 41.3 |
| CL [17] | ECCV18 | √ | √ | 132.0 | 191.0 | - | - | - | - |
| CSRNet [23] | CVPR18 | √ | √ | - | - | 68.2 | 115.0 | 10.6 | 16.0 |
| L2R [31] | TPAMI19 | √ | √ | 124.0 | 196.0 | 73.6 | 112.0 | 13.7 | 21.4 |
| CFF [42] | ICCV19 | √ | √ | - | - | 65.2 | 109.4 | 7.2 | 12.2 |
| PGCNet [64] | ICCV19 | √ | √ | - | - | 57.0 | **86.0** | 8.8 | 13.7 |
| TEDnet [18] | CVPR19 | √ | √ | 113.0 | 188.0 | 64.2 | 109.1 | 8.2 | 12.8 |
| CAN [30] | CVPR19 | √ | √ | 107.0 | 183.0 | 62.3 | 100.0 | 7.8 | 12.2 |
| S-DCNet [60] | ICCV19 | √ | √ | 104.4 | 176.1 | 58.3 | 95.0 | **6.7** | 10.7 |
| DSSI-Net [26] | ICCV19 | √ | √ | 99.1 | 159.2 | 60.6 | 96.0 | 6.8 | **10.3** |
| BL [34] | ICCV19 | √ | √ | 88.7 | 154.8 | 62.8 | 101.8 | 7.7 | 12.7 |
| ASNet [19] | CVPR20 | √ | √ | 91.5 | 159.7 | 57.7 | 90.1 | - | - |
| LibraNet [25] | ECCV20 | √ | √ | 88.1 | **143.7** | **55.9** | 97.1 | 7.3 | 11.3 |
| NoisyCC [51] | NeurIPS20 | √ | √ | 85.8 | 150.6 | 61.9 | 99.6 | 7.4 | 11.3 |
| DM-Count [51] | NeurIPS20 | √ | √ | **85.6** | 148.3 | 59.7 | 95.7 | 7.4 | 11.8 |
| Method in [67]* | ECCV20 | – | √ | - | - | 104.6 | 145.2 | 12.3 | 21.2 |
| MATT [21]* | 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** |

Table 1: Quantitative comparison (in terms of MAE and MSE) of the proposed method and some popular methods on three widely adopted benchmark datasets. * represents the weakly-supervised method.

| Method | Year | Training label | | Val set | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Low | | Medium | | High | | Overall | |
| | | Location | Crowd number | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE |
| MCNN [70] | CVPR16 | √ | √ | 90.6 | 202.9 | 125.3 | 259.5 | 494.9 | 856.0 | 160.6 | 377.7 |
| CMTL [45] | AVSS17 | √ | √ | 50.2 | 129.2 | 88.1 | 170.7 | 583.1 | 986.5 | 138.1 | 379.5 |
| DSSI-Net [26] | ICCV19 | √ | √ | 50.3 | 85.9 | 82.4 | 164.5 | 436.6 | 814.0 | 116.6 | 317.4 |
| CAN [30] | CVPR19 | √ | √ | 34.2 | 69.5 | 65.6 | 115.3 | 336.4 | **619.7** | 89.5 | 239.3 |
| SANet [3] | ECCV18 | √ | √ | 13.6 | 26.8 | 50.4 | 78.0 | 397.8 | 749.2 | 82.1 | 272.6 |
| CSRNet [23] | CVPR18 | √ | √ | 22.2 | 40.0 | 49.0 | 99.5 | 302.5 | 669.5 | 72.2 | 249.9 |
| CG-DRCN [44] | PAMI20 | √ | √ | 17.1 | 44.7 | 40.8 | **71.2** | 317.4 | 719.8 | 67.9 | 262.1 |
| MBTTBF [47] | ICCV19 | √ | √ | 23.3 | 48.5 | 53.2 | 119.9 | 294.5 | 674.5 | 73.8 | 256.8 |
| SFCN [57] | CVPR19 | √ | √ | 11.8 | 19.8 | **39.3** | 73.4 | 297.3 | 679.4 | 62.9 | 247.5 |
| BL [34] | ICCV19 | √ | √ | **6.9** | **10.3** | 39.7 | 85.2 | **279.8** | 620.4 | **59.3** | **229.2** |
| **TransCrowd-Token (our)*** | - | - | √ | 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** |

Table 2: Quantitative results on the JHU-Crowd++ (val set) dataset. ”Low”, ”Medium” and ”High” respectively indicates three categories based on different ranges:[0,50], (50,500], and >500. * represents the weakly-supervised crowd counting methods.

# 6. Analysis

## 6.1. The input of regression head

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 [10] and ViT [11]. TransCrowd-GAP utilizes global average pooling to obtain the pooled visual tokens for count prediction. The result of TransCrowd-Token and TransCrowd-GAP are listed in Tab. 1, Tab. 2, Tab. 3, and Tab. 4. 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 Fig. 4 to understand what the Transformer attends to. We observe that both TransCrowd-Token and TransCrowd-GAP can successfully

| Method | Year | Training label | | Testing set | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Low | | Medium | | High | | Overall | |
| | | Location | Crowd number | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE |
| MCNN [70] | CVPR16 | √ | √ | 97.1 | 192.3 | 121.4 | 191.3 | 618.6 | 1,166.7 | 188.9 | 483.4 |
| CMTL [45] | AVSS17 | √ | √ | 58.5 | 136.4 | 81.7 | 144.7 | 635.3 | 1,225.3 | 157.8 | 490.4 |
| DSSI-Net [26] | ICCV19 | √ | √ | 53.6 | 112.8 | 70.3 | 108.6 | 525.5 | 1,047.4 | 133.5 | 416.5 |
| CAN [30] | CVPR19 | √ | √ | 37.6 | 78.8 | 56.4 | 86.2 | 384.2 | 789.0 | 100.1 | 314.0 |
| SANet [3] | ECCV18 | √ | √ | 17.3 | 37.9 | 46.8 | 69.1 | 397.9 | 817.7 | 91.1 | 320.4 |
| CSRNet [23] | CVPR18 | √ | √ | 27.1 | 64.9 | 43.9 | 71.2 | 356.2 | 784.4 | 85.9 | 309.2 |
| CG-DRCN [44] | PAMI20 | √ | √ | 19.5 | 58.7 | 38.4 | 62.7 | 367.3 | 837.5 | 82.3 | 328.0 |
| MBTTBF [47] | ICCV19 | √ | √ | 19.2 | 58.8 | 41.6 | 66.0 | 352.2 | 760.4 | 81.8 | 299.1 |
| SFCN [57] | CVPR19 | √ | √ | 16.5 | 55.7 | 38.1 | 59.8 | **341.8** | **758.8** | 77.5 | **297.6** |
| BL [34] | ICCV19 | √ | √ | **10.1** | **32.7** | **34.2** | **54.5** | 352.0 | 768.7 | **75.0** | 299.9 |
| **TransCrowd-Token (our)*** | - | - | √ | 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** |

Table 3: Quantitative results on the JHU-Crowd++ (testing set) dataset. ”Low”, ”Medium” and ”High” respectively indicates three categories based on different ranges:[0,50], (50,500], and >500. * represents the weakly-supervised crowd counting methods.

| Method | Year | Training label | | Val set | | Testing set | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Overall | | Overall | | Scene Level (only MAE) | |
| | | Location | Crowd number | MAE | MSE | MAE | MSE | Avg. | $S0 \sim S4$ |
| C3F-VGG [13] | Tech19 | √ | √ | 105.79 | 504.39 | 127.0 | 439.6 | 666.9 | 140.9/26.5/58.0/307.1/2801.8 |
| CSRNet [23] | CVPR18 | √ | √ | 104.89 | 433.48 | 121.3 | 387.8 | 522.7 | 176.0/35.8/59.8/285.8/2055.8 |
| PCC-Net-VGG [14] | CVPR19 | √ | √ | 100.77 | 573.19 | 112.3 | 457.0 | 777.6 | 103.9/13.7/42.0/259.5/3469.1 |
| CAN [30] | CVPR19 | √ | √ | 93.58 | 489.90 | 106.3 | **386.5** | 612.2 | 82.6/14.7/46.6/269.7/2647.0 |
| SFCN† [57] | CVPR19 | √ | √ | 95.46 | 608.32 | 105.7 | 424.1 | 712.7 | 54.2/14.8/44.4/249.6/3200.5 |
| BL [34] | ICCV19 | √ | √ | 93.64 | 470.38 | 105.4 | 454.2 | 750.5 | 66.5/8.7/41.2/249.9/3386.4 |
| KDMG [53] | PAMI20 | √ | √ | - | - | 100.5 | 415.5 | 632.7 | 77.3/10.3/38.5/259.4/2777.9 |
| NoisyCC [51] | NeurIPS20 | √ | √ | - | - | 96.9 | 534.2 | 608.1 | 218.7/10.7/35.2/203.2/2572.8 |
| DM-Count [54] | NeurIPS20 | √ | √ | **70.5** | **357.6** | **88.4** | 388.6 | **498.0** | 146.6/7.6/31.2/228.7/2075.8 |
| **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 |

Table 4: Comparison of the counting performance on the NWPU-Crowd. $S0 \sim S4$ respectively indicate five categories according to the different number range: 0, $(0, 100]$, $(100, 500]$, $(500, 5000]$, >5000. * represents the weakly-supervised crowd counting methods.

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 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 [23]) and the proposed TransCrowd. Detailed convergence curves are shown in Fig. 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 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 CSR-Net is oscillating. These observations show the potential value of the Transformer in the counting task.

### 6.4. Comparison of Different Pre-trained strategies.

In this section, we study the impact of the pre-trained model in TransCrowd. We choose the popular CNN-based method CSRNet [23] as a comparison, and the results are listed in Tab. 5. 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-tune on ShanghaiTech part A. (3) **Pre-GCC**: The models are pre-trained on GCC [57], a synthetic dataset, and are fine-tuned on ShangahiTech part A dataset.

| Method | Year | Training label | | None | | Pre-ImgNet | | Pre-GCC | |
|---|---|---|---|---|---|---|---|---|---|
| | | Location | Crowd number | MAE | MSE | MAE | MSE | MAE | MSE |
| CSRNet [34] | 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 5: 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 | | 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 [70] | CVPR16 | √ | √ | 221.4 | 357.8 | 85.2 | 142.3 | - | - | - | - |
| D-ConvNet [43] | ECCV18 | √ | √ | 140.4 | 226.1 | 49.1 | 99.2 | - | - | - | - |
| RRSP[52] | CVPR19 | √ | √ | - | - | 40.0 | 68.5 | - | - | - | - |
| BL [34] | 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** |

Table 6: Experimental results on the transferability of different methods under cross-dataset evaluation.

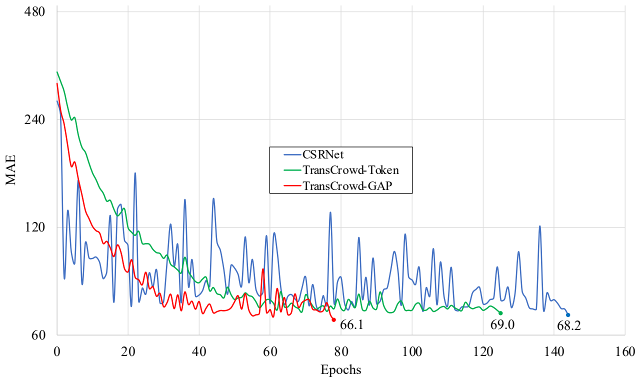


Figure 5: Convergence curves of CSRNet, TransCrowd-Token, and TransCrowd-GAP on ShanghaiTech part A dataset. The proposed TransCrowd-GAP achieves the best counting performance and is fast-converging.

From the Tab. 5, 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 pre-trained on the GCC dataset, the proposed method can even outperform several recent fully-supervised methods (e.g., CFF [42], TEDNet [18]). 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.

### 6.5. 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 Tab. 6. Although our method is a weakly-supervised paradigm, we still achieve highly competitive performance, which shows remarkable transferability.

## 7. Conclusion

In this work, 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.

## References

[1] Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. Localization in the crowd with topological constraints. In *AAAI*, 2021. 2

[2] Shuai Bai, Zhiqun He, Yu Qiao, Hanzhe Hu, Wei Wu, and Junjie Yan. Adaptive dilated network with self-correction

supervision for counting. In *CVPR*, pages 4594–4603, 2020. 1

[3] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. Scale aggregation network for accurate and efficient crowd counting. In *ECCV*, pages 734–750, 2018. 7, 8

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 2, 4

[5] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. *arXiv preprint arXiv:2012.00364*, 2020. 4

[6] Jieneng Chen, Yongyi Lu, Qihang Yu, Xiangde Luo, Ehsan Adeli, Yan Wang, Le Lu, Alan L Yuille, and Yuyin Zhou. Transunet: Transformers make strong encoders for medical image segmentation. *arXiv preprint arXiv:2102.04306*, 2021. 4

[7] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. *arXiv preprint arXiv:2103.15436*, 2021. 4

[8] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 4

[9] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. *arXiv preprint arXiv:2011.09094*, 2020. 4

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 3, 4, 7

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3, 4, 5, 7

[12] Dawei Du, Longyin Wen, Pengfei Zhu, Heng Fan, Qinghua Hu, Haibin Ling, Mubarak Shah, Junwen Pan, Ali Al-Ali, Amr Mohamed, et al. Visdrone-cc2020: The vision meets drone crowd counting challenge results. In *European Conference on Computer Vision*, pages 675–691. Springer, 2020. 2

[13] Junyu Gao, Wei Lin, Bin Zhao, Dong Wang, Chenyu Gao, and Jun Wen. Cˆ 3 framework: An open-source pytorch code for crowd counting. *arXiv preprint arXiv:1907.02724*, 2019. 8

[14] Junyu Gao, Qi Wang, and Xuelong Li. Pcc net: Perspective crowd counting via spatial convolutional network. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(10):3486–3498, 2019. 3, 8

[15] Bin Guo, Zhu Wang, Zhiwen Yu, Yu Wang, Neil Y Yen, Runhe Huang, and Xingshe Zhou. Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm. *ACM computing surveys (CSUR)*, 48(1):1–31, 2015. 1

[16] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. 2016. 4

[17] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. In *ECCV*, pages 532–546, 2018. 5, 7

[18] Xiaolong Jiang, Zehao Xiao, Baochang Zhang, Xiantong Zhen, Xianbin Cao, David Doermann, and Ling Shao. Crowd counting and density estimation by trellis encoder-decoder networks. In *CVPR*, pages 6133–6142, 2019. 2, 7, 9

[19] Xiaoheng Jiang, Li Zhang, Mingliang Xu, Tianzhu Zhang, Pei Lv, Bing Zhou, Xin Yang, and Yanwei Pang. Attention scaling for crowd counting. In *CVPR*, pages 4706–4715, 2020. 2, 3, 7

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[21] Yinjie Lei, Yan Liu, Pingping Zhang, and Lingqiao Liu. Towards using count-level weak supervision for crowd counting. *Pattern Recognition*, 109:107616, 2021. 1, 2, 3, 5, 7

[22] Wenhao Li, Hong Liu, Runwei Ding, Mengyuan Liu, and Pichao Wang. Lifting transformer for 3d human pose estimation in video. *arXiv preprint arXiv:2103.14304*, 2021. 4

[23] Yuhong Li, Xiaofan Zhang, and Deming Chen. Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In *CVPR*, pages 1091–1100, 2018. 1, 2, 6, 7, 8

[24] Dingkang Liang, Wei Xu, Yingying Zhu, and Yu Zhou. Focal inverse distance transform maps for crowd localization and counting in dense crowd. *arXiv preprint arXiv:2102.07925*, 2021. 2

[25] Liang Liu, Hao Lu, Hongwei Zou, Haipeng Xiong, Zhiguo Cao, and Chunhua Shen. Weighing counts: Sequential crowd counting by reinforcement learning. In *European Conference on Computer Vision*, pages 164–181. Springer, 2020. 7

[26] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. Crowd counting with deep structured scale integration network. In *ICCV*, pages 1774–1783, 2019. 7, 8

[27] Ning Liu, Yongchao Long, Changqing Zou, Qun Niu, Li Pan, and Hefeng Wu. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In *CVPR*, pages 3225–3234, 2019. 1, 3

[28] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3694–3702, 2021. 4

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[30] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. Context-aware crowd counting. In *CVPR*, pages 5099–5108, 2019. 7, 8

[31] Xialei Liu, Joost Van De Weijer, and Andrew D Bagdanov. Exploiting unlabeled data in cnns by self-supervised learning to rank. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1862–1878, 2019. 2, 3, 6, 7

[32] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019. 3

[33] Yuting Liu, Miaojing Shi, Qijun Zhao, and Xiaofang Wang. Point in, box out: Beyond counting persons in crowds. In *CVPR*, 2019. 2

[34] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *ICCV*, pages 6142–6151, 2019. 2, 6, 7, 8, 9

[35] Weian Mao, Yongtao Ge, Chunhua Shen, Zhi Tian, Xinlong Wang, and Zhibin Wang. Tfpose: Direct human pose estimation with transformers. *arXiv preprint arXiv:2103.15320*, 2021. 4

[36] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 3

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2

[38] Deepak Babu Sam, Skand Vishwanath Peri, Mukuntha Narayanan Sundararaman, Amogh Kamath, and Venkatesh Babu Radhakrishnan. Locate, size and count: Accurately resolving people in dense crowds via detection. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[39] Deepak Babu Sam, Neeraj N Sajjan, Himanshu Maurya, and R Venkatesh Babu. Almost unsupervised learning for dense crowd counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8868–8875, 2019. 3

[40] Xiang Sheng, Jian Tang, Xuejie Xiao, and Guoliang Xue. Leveraging gps-less sensing scheduling for green mobile crowd sensing. *IEEE Internet of Things Journal*, 1(4):328–336, 2014. 1

[41] Miaojing Shi, Zhaohui Yang, Chao Xu, and Qijun Chen. Revisiting perspective information for efficient crowd counting. In *CVPR*, pages 7279–7288, 2019. 3

[42] Zenglin Shi, Pascal Mettes, and Cees GM Snoek. Counting with focus for free. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4200–4209, 2019. 7, 9

[43] Zenglin Shi, Le Zhang, Yun Liu, Xiaofeng Cao, Yangdong Ye, Ming-Ming Cheng, and Guoyan Zheng. Crowd counting with deep negative correlation learning. In *CVPR*, pages 5382–5390, 2018. 9

[44] Vishwanath Sindagi, Rajeev Yasarla, and Vishal MM Patel. Jhu-crowd++: Large-scale crowd counting dataset and a benchmark method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 5, 7, 8

[45] Vishwanath A Sindagi and Vishal M Patel. Cnn-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *AVSS*, pages 1–6, 2017. 7, 8

[46] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV*, pages 1861–1870, 2017. 2

[47] Vishwanath A Sindagi and Vishal M Patel. Multi-level bottom-top and top-bottom feature fusion for crowd counting. In *ICCV*, pages 1002–1012, 2019. 7, 8

[48] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M Patel. Medical transformer: Gated axial-attention for medical image segmentation. *arXiv preprint arXiv:2102.10662*, 2021. 4

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 3, 4

[50] Matthias von Borstel, Melih Kandemir, Philip Schmidt, Madhavi K Rao, Kumar Rajamani, and Fred A Hamprecht. Gaussian process density counting from weak supervision. In *European Conference on Computer Vision*, pages 365–380. Springer, 2016. 3

[51] Jia Wan and Antoni Chan. Modeling noisy annotations for crowd counting. *Advances in Neural Information Processing Systems*, 33, 2020. 7, 8

[52] Jia Wan, Wenhan Luo, Baoyuan Wu, Antoni B Chan, and Wei Liu. Residual regression with semantic prior for crowd counting. In *CVPR*, pages 4036–4045, 2019. 9

[53] Jia Wan, Qingzhong Wang, and Antoni B Chan. Kernel-based density map generation for dense object counting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 8

[54] Boyu Wang, Huidong Liu, Dimitris Samaras, and Minh Hoai. Distribution matching for crowd counting. *arXiv preprint arXiv:2009.13077*, 2020. 8

[55] Ning Wang, Wengang Zhou, Jie Wang, and Houqaing Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. *arXiv preprint arXiv:2103.11681*, 2021. 4

[56] Qi Wang, Junyu Gao, Wei Lin, and Xuelong Li. Nwpu-crowd: A large-scale benchmark for crowd counting and localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 5

[57] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. In *CVPR*, pages 8198–8207, 2019. 2, 3, 7, 8

[58] Xinpeng Wang, Chandan Yeshwanth, and Matthias Nießner. Sceneformer: Indoor scene generation with transformers. *arXiv preprint arXiv:2012.09793*, 2020. 4

[59] Bichen Wu, Chenfeng Xu, Xiaoliang Dai, Alvin Wan, Peizhao Zhang, Masayoshi Tomizuka, Kurt Keutzer, and Peter Vajda. Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, 2020. 4

[60] Haipeng Xiong, Hao Lu, Chengxin Liu, Liang Liu, Zhiguo Cao, and Chunhua Shen. From open set to closed set: Counting objects by spatial divide-and-conquer. In *ICCV*, pages 8362–8371, 2019. 7

[61] Chenfeng Xu, Dingkang Liang, Yongchao Xu, Song Bai, Wei Zhan, Xiang Bai, and Masayoshi Tomizuka. Autoscale: learning to scale for crowd counting. *arXiv preprint arXiv:1912.09632*, 2019. 2

[62] Chenfeng Xu, Kai Qiu, Jianlong Fu, Song Bai, Yongchao Xu, and Xiang Bai. Learn to scale: Generating multipolar normalized density maps for crowd counting. In *ICCV*, pages 8382–8390, 2019. 1, 2

[63] Wei Xu, Dingkang Liang, Yixiao Zheng, and Zhanyu Ma. Dilated-scale-aware attention convnet for multi-class object counting. *arXiv preprint arXiv:2012.08149*, 2020. 2

[64] Zhaoyi Yan, Yuchen Yuan, Wangmeng Zuo, Xiao Tan, Yezhen Wang, Shilei Wen, and Errui Ding. Perspective-guided convolution networks for crowd counting. In *ICCV*, pages 952–961, 2019. 3, 7

[65] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Towards explainable human pose estimation by transformer. *arXiv preprint arXiv:2012.14214*, 2020. 4

[66] Yifan Yang, Guorong Li, Zhe Wu, Li Su, Qingming Huang, and Nicu Sebe. Reverse perspective network for perspective-aware object counting. In *CVPR*, pages 4374–4383, 2020. 3

[67] Yifan Yang, Guorong Li, Zhe Wu, Li Su, Qingming Huang, and Nicu Sebe. Weakly-supervised crowd counting learns from sorting rather than locations. In *The European Conference on Computer Vision*. Springer, 2020. 1, 2, 3, 5, 7

[68] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019. 3

[69] Anran Zhang, Lei Yue, Jiayi Shen, Fan Zhu, Xiantong Zhen, Xianbin Cao, and Ling Shao. Attentional neural fields for crowd counting. In *ICCV*, pages 5714–5723, 2019. 2, 3

[70] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. Single-image crowd counting via multi-column convolutional neural network. In *CVPR*, pages 589–597, 2016. 1, 2, 5, 6, 7, 8, 9

[71] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *arXiv preprint arXiv:2012.15840*, 2020. 2, 4

[72] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 4