Rethinking Global Context in Crowd Counting

Guolei Sun[1], Yun Liu[†2], Thomas Probst[3], Danda Pani Paudel[1], Nikola Popovic[1] and Luc Van Gool[1]

[1]Computer Vision Lab, ETH Zürich, Zürich, Switzerland.
[2]Institute for Infocomm Research, A*STAR, Singapore.
[3]Magic Leap, Zurich, Switzerland.

Abstract

This paper investigates the role of global context for crowd counting. Specifically, a pure transformer is used to extract features with global information from overlapping image patches. Inspired by classification, we add a context token to the input sequence, to facilitate information exchange with tokens corresponding to image patches throughout transformer layers. Due to the fact that transformers do not explicitly model the tried-and-true channel-wise interactions, we propose a token-attention module (TAM) to recalibrate encoded features through channel-wise attention informed by the context token. Beyond that, it is adopted to predict the total person count of the image through regression-token module (RTM). Extensive experiments on various datasets, including ShanghaiTech, UCF-QNRF, JHU-CROWD++ and NWPU, demonstrate that the proposed context extraction techniques can significantly improve the performance over the baselines.

Keywords: crowd counting, vision transformer, global context, attention, density map

1 Introduction

At first sight, counting the size of a crowd present in an image is equivalent to the problem of detecting and counting of person instances [1, 2]. Such direct approaches however have been shown not to perform well, because generic detectors suffer from the small instance size and severe occlusions present in crowded regions [3, 4] – typically a person covers only a small number of pixels, and only few body parts are visible (often just the head) [5]. State-of-the-art crowd counting approaches therefore rely on the prediction of crowd density maps, a localized, pixel-wise measure of person presence [3, 5–29].

To this end, underlying network architectures need to integrate context across location and scales [3, 11, 30]. This is crucial due to the vast variety of possible appearances of a given crowd density. In other words, the ability to integrate a large context makes it possible to adapt the density estimation to an expectation raised by the given scene, beyond the tunnel vision of local estimation. *Geometry* and *semantics* are two of the main aspects of scene context [31, 32], that can serve this goal for crowd counting [31, 33]. Unfortunately, even if we manage to model and represent such knowledge, it is very cumbersome to obtain, and therefore not practical for many applications of image-based crowd counting. This also reflects the setup of the most popular crowd counting challenge datasets considered in this paper [5, 7, 34, 35].

† Corresponding authors.

On the bright side, even in the absence of such direct knowledge, we can benefit from the recent progress in geometric and semantic learning on a conceptual level – by studying the inductive biases. In fact, the development of computer vision in the last decade demonstrated the possibility to implicitly learn representations capturing rich geometric [36] and semantic [37, 38] information from a single image. Recently, the advantageous nature of global interaction over convolutional neural networks (CNNs) has been demonstrated for both geometric features for monocular depth prediction [39], as well as for semantic features in segmentation [37, 40]. The aforementioned works attribute the success of the transformer [41, 42] to global receptive fields, which has been a bottleneck in previous CNN-based approaches. Moreover, CNNs by design apply the same operation on all locations, rendering it a sub-optimal choice for exploiting information about the geometric and semantic composition of the scene.

As geometric and semantic understanding are crucial aspects of scene context for the task of crowd counting, we hypothesize that superior capabilities of transformers on these aspects are also indicative of a more suitable inductive bias for crowd counting. To investigate our hypothesis, we adapt the vision transformers [37, 42, 43] for the task of crowd counting.

Unlike image classification [42], crowd counting is a dense prediction task. Following our previous discussion, the learning of crowd counting is also predicated on the global context of the image. To capture both spatial information for dense prediction, as well as the necessary scene context, we maintain both local tokens (representing image patches) and a context token (representing image context). We then introduce a token attention module (TAM) to refine the encoded features informed by the context token. We further guide the learning of the context token by using a regression token module (RTM), that accommodates an auxiliary loss on the regression of the total count of the crowd. Following [37], the refined transformer output is then mapped to the desired crowd density map using two deconvolution layers. Please refer to Fig. 1 for an illustration of the overall framework.

In particular, our proposed TAM is designed to address the observation that the multi-head self-attention (MHSA) in vision transformers only models spatial interactions, while the tried-and-true channel-wise interactions have also been proved to be of vital effectiveness [44, 45]. To this end, TAM imprints the context token on the local tokens by conditional recalibration of feature channels, therefore explicitly modelling channel-wise interdependencies. Current widely-used methods to achieve this goal includes SENet [44] and CBAM [45]. They use simple aggregation technique such as global average pooling or global maximum pooling on the input features to obtain channel-wise statistics (global abstraction), which are then used to capture channel-wise dependencies. For transformers, we propose a natural and elegant way to model channel relationships by extending the input sequence with a context token and introducing the TAM to recalibrate local tokens through channel-wise attention informed by the context token. The additional attention across feature channels further facilitates the learning of global context.

We also adopt context token which interacts with other patch tokens throughout the transformers to regress the total crowd count of the whole image. This is achieved by the proposed RTM, containing a two-layer MLP. On the one hand, the syzygy of TAM and RTM forces the context token to collect and distribute image-level count estimates from and to all local tokens, leading to a better representation of context token. On the other hand, it helps to learn better underlying features for the task and reduce overfitting within the network, similar to *auxiliary-task learning* [46].

In summary, we provide another perspective on density-supervised crowd counting, through the lens of learning features with global context. Specifically, we introduce a context token tasked with the refinement of local feature tokens through a novel framework of token-attention and regression-token modules. Our framework thereby addresses the shortcomings of CNNs with regards to capturing global context for the problem of crowd counting. We conduct experiments on various popular datasets, including ShanghaiTech, UCF-QNRF, JHU-CROWD++ and NWPU. The experimental results demonstrate that the proposed context extraction techniques can significantly improve the performance over the

baselines and thus open a new path for crowd counting.

2 Related works

2.1 Crowd Counting

Most crowd counting methods are based on convolutional neural networks (CNNs), which can be divided into three categories: counting by regression [47, 48], counting by detection [1, 2, 49], and counting by estimating density maps [3, 5–29, 50–52]. The regression-based methods directly regress the total count of the crowd in the image, while the location of the people is not considered. Detection-based approaches first detect the people and then count the number of detections. However, those methods do not perform well in many interesting cases, where detection is difficult due to occlusions and high density of people. As a consequence, the mainstream direction of crowd counting is to estimate the density map of the image and then sum over the density map to obtain the total count. For this work, we also follow the direction of estimating density map. Different from existing methods, we target the crowd counting problem from the perspective of global information.

The methods [3, 6, 11] which exploit large receptive field for crowd counting have been proposed. The techniques include: using spatial average pooling [3] or dilation convolution [11], and increasing network depth [6]. However, the receptive field is still limited, rather than global. Technically, only local information is used. In this work, we propose to use global information for crowd counting, by taking advantage of recent transformer technique. To the best of our knowledge, there are only limited works [53] adopting vision transformers to conduct crowd counting. However, the method of [53] is concerned with weakly supervised crowd counting in the sense of only regressing the total count, where dot annotations are not available. Its performance therefore cannot compete with the mainstream point-supervised crowd counting methods on most standard benchmarks [5, 7, 35]. Differently, we investigate point-supervised crowd counting using vision transformers and show the effectiveness of global context in crowd counting.

2.2 Vision Transformer

The transformer, relying on self-attention mechanism [41], was first introduced in natural language processing [41], and has been dominating this area ever since. In general, a transformer contains a MHSA module and a multi-layer perceptron (MLP), to model the contextual information within input sequences through global interaction. Recently, pioneer works such as ViT [42] and DETR [54] utilize transformers to solve vision problems. Transformers have shown to be effective in tasks of image classification [42], object detection [54], and semantic/instance segmentation [37]. However, the exploration of transformers for crowd counting [53] has been limited. In this paper, we demonstrate the power of transformers in point-supervised crowd counting setup, where persons are represented with a dense map.

3 Method

3.1 Problem Definition

Given a training dataset of images $\{\boldsymbol{I}_i\}_K \subseteq \mathbb{R}^{c\times h\times w}$ and crowd density label maps $\{\boldsymbol{D}_i\}_K \subseteq \mathbb{R}^{h\times w}$, our goal is to learn a neural network model $\mathcal{M}: \mathbb{R}^{c\times h\times w} \rightarrow \mathbb{R}^{h\times w}$, that estimates the crowd density map $\boldsymbol{D}^{'} = \mathcal{M}(\boldsymbol{I})$ and therefore counts the number of visible people $\|\boldsymbol{D}^{'}\|_1$ from an unseen image $\boldsymbol{I}$.

3.2 Transformer-based Crowd Counting

Most crowd counting methods in the literature that consider crowd counting as a dense prediction task are based on CNNs [3, 11, 13]. Since CNN-based encoders can only exploit the local information within the fix-sized window, some approaches are proposed to increase the receptive fields, by dilated convolutions [11] or using deeper networks [15]. In this section, we present our transformer-based approach for crowd counting, which is designed to overcome this limitation by explicitly modelling global context. Our presentation follows the data flow of our framework as depicted in Fig. 1.

Overlapping Split. In the seminal ViT [42], the input image is split into non-overlapping

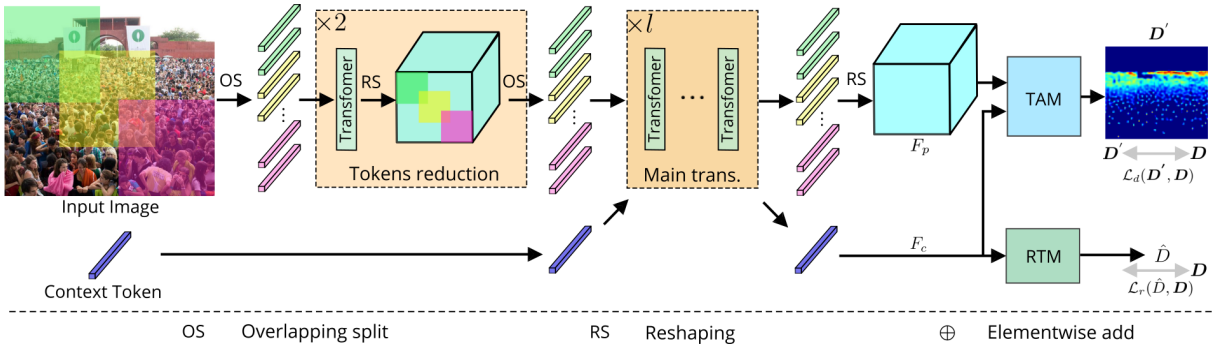

Fig. 1: **Network Overview.** The input image is first split into overlapping patches. Then, those patches go through tokens reduction block and main transformer to learn features with global information. To abstract global information, context token (blue vector) is added to the input sequence before the main transformer. The encoded features are processed by TAM and regression-token module (RTM). The small decoder after TAM is not shown for simplicity.

patches, leading to the problem that the local structure around the patches is destroyed. Instead, we split the input into overlapping patches, following [43]. The process of overlapping split is similar to an convolution operation and the patch size of $k \times k$ is similar to the kernel size. Specifically, the input image $\boldsymbol{I}$ is first padded by p pixels on each side. The overlapping patches are obtained by moving the patch window $(k \times k)$ across the whole image with stride s $(s < k)$. Each patch has $k \times k \times c$ elements, which are flattened to $\mathbb{R}^{ck^2}$. The length of patches is given by

$$N_0 = h_0 \times w_0, \tag{1}$$

where $h_0 = \lfloor \frac{h+2p-k}{s} + 1 \rfloor$ and $w_0 = \lfloor \frac{w+2p-k}{s} + 1 \rfloor$. After concatenating all patches together, image tokens are obtained, denoted by $Z_0 \in \mathbb{R}^{N_0 \times ck^2}$. Later, we process Z_0 by the tokens reduction block, followed by the main transformer.

Tokens Reduction. We first input Z_0 to a transformer layer and obtain Z_1, formulated as

$$Z_1 = \text{MLP}(\text{MHSA}(Z_0)), \tag{2}$$

where $Z_1 \in \mathbb{R}^{N_0 \times d}$, and d is the dimension of *query*, *key*, and *value*. Since the sequence length N_0 is relatively large due to the overlapping split, we reshape Z_1 back to $\mathbb{R}^{h_0 \times w_0 \times d}$ and perform overlapping split again to reduce the spatial size by stride s. Let $Z_1^{'}$ be the obtained tokens with size of $\mathbb{R}^{N_1 \times dk^2}$, and $N_1 = h_1 \times w_1$, where $h_1 = \lfloor \frac{h_0+2p-k}{s} + 1 \rfloor$ and $w_1 = \lfloor \frac{w_0+2p-k}{s} + 1 \rfloor$. Following [43], this process is repeated twice and we obtain $Z_2^{'} \in \mathbb{R}^{N_2 \times dk^2}$, where $N_2 = h_2 \times w_2$. The length of sequence N_2 is thereby reduced to a manageable scale. Since the dependency among those pixels around the original non-overlapping split (as in ViT [42]) is well-modelled, we fix the length of sequence as $N = N_2$ and do not reduce it further, in order to maintain both the representation capability and efficiency. After projecting $Z_2^{'}$ to $T \in \mathbb{R}^{N \times d}$, we process T by deep-narrow ViT [42].

Context Token. Recall that we approach crowd counting as a dense prediction problem, and each patch token transforms local RGB input to a local density map prediction. Therefore, even though the patch tokens T are in principle able to interact globally in ViT [42], our mode of dense supervision renders each token to be primarily concerned with its local region. In order to foster global information exchange without compromising capacity for local features, we delegate the collection of global context to a context token t_{con}. In contrast, previous transformer-based approaches to dense prediction [37] only employ local tokens without explicitly modelling the global context. In our framework, the context token is the key input for the TAM as described in §3.3, which disseminates the global context back to the local tokens. The local tokens therefore remain dedicated to their local predictions. In §3.4 we explain how to guide the learning of context token through the RTM module. But first, we give a brief description of the main transformer of our framework.

Main Transformer. The main transformer follow the same architecture as ViT [42], but have less channels in intermediate layers to reduce redundancy within original ViT model. As laid out above, we append the context token t_{con} to the patch tokens T to facilitate global-local interaction. Following [42], position embedding E is also added. The main transformer is denoted as follows:

$$\begin{aligned} T_0 &= [T;\ t_{con}] + E, \quad E \in \mathbb{R}^{(N+1)\times d}, \\ T_i^{'} &= \mathrm{MHSA}(T_{i-1}) + T_{i-1}, \quad i = 1, ..., l, \\ T_i &= \mathrm{MLP}(T_i^{'}) + T_i^{'}, \quad i = 1, ..., l. \end{aligned} \tag{3}$$

Here, l is the number of layers in the main transformer. T_l is the feature sequence from the last layer of transformers. It has global receptive fields which are effective for crowd counting task. Since a context token is added in the beginning, we split T_l as follows

$$F_p = T_l[:N], \quad F_c = T_l[N], \tag{4}$$

where $F_p \in \mathbb{R}^{N\times d}$ is the feature corresponding to image patches, and $F_c \in \mathbb{R}^d$ is the feature vector corresponding to context token t_{con}. To recover spatial structure, F_p is reshaped to $\mathbb{R}^{d\times h_2\times w_2}$. F_p is further refined by TAM to predict the density map and F_c is used by the proposed regression-token module (RTM) to predict the overall count for the image .

3.3 Token-attention Module (TAM)

The task of the TAM is to refine the local feature map F_p used to predict the crowd density map, conditioned upon the context token feature F_c. This will infuse the global context information into the local density predictions. Before presenting the details of TAM, we give a brief analysis of the preceding transformer layers to motivate the proposed mechanism.

Spatial and Channel Attention. Recall that the token T_l is produced from T_{l-1} by the last transformer layer, which performs the operations

$$\begin{aligned} T_l^{'} &= \mathrm{softmax}(\frac{T_{l-1}\boldsymbol{W}_Q(T_{l-1}\boldsymbol{W}_K)^T}{\sqrt{d}})T_{l-1}\boldsymbol{W}_V + T_{l-1}, \\ T_l &= \mathrm{MLP}(T_l^{'}) + T_l^{'}, \end{aligned} \tag{5}$$

where $T_{l-1}/T_l^{'}/T_l \in \mathbb{R}^{N\times d}$, and $\boldsymbol{W}_Q/\boldsymbol{W}_K/\boldsymbol{W}_V \in \mathbb{R}^{d\times d}$ are the learnable parameters for generating $(query, key, value)$. For simplicity of notation, MHSA is represented by a special case where a single self-attention (SA) operation is performed. We can see that a token $T_l^{'}[i]$ (corresponding to a specific image patch or the context token), is generated by a weighted summation of tokens T_{l-1}. Therefore, transformers are inherently equipped with spatial attention mechanism which pays more attention to the relevant spatial regions (tokens). However, the feature channel interdependencies are not explicitly modelled in the transformer operations (5). Explicitly modelling channel relationships, so that the network has the capability to focus on important feature channels, leads to enhanced features [44, 45]. This is also confirmed by our experiments: while no improvements are obtained by adding spatial attention, introducing channel attention yields better predictions. To this end, we introduce TAM as a mechanism to perform feature channel attention.

Global Abstraction. Global abstraction is used to provide cues for channel interdependencies. In SE, global abstraction is obtained by conducting global average pooling across spatial dimensions on the input itself, while CBAM [45] merges both global average pooling and global maximum pooling. For transformers, we propose a natural and elegant approach to abstract global information, by extending the input sequence with a context token, as introduced above. Since the obtained feature F_c from context token has a global overview throughout transformer layers, we adopt it to provide information on which channels are important for predicting density map. The comparison between SE and TAM is shown in Fig. 2, where the sigmoid is omitted for simplicity. The superiority of the proposed TAM over SE [44] and CBAM [45] is validated in §4.3.

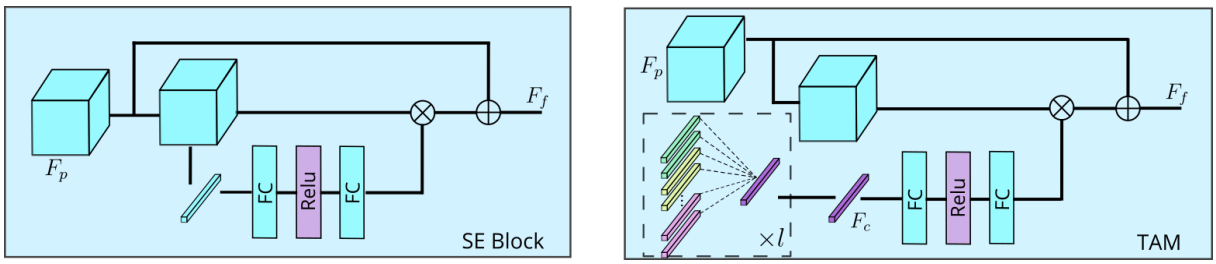

Fig. 2: **Comparison between SE block [44] and TAM.** Different from SE block which obtains global information from input features, TAM adopts context token feature to provide channel relations.

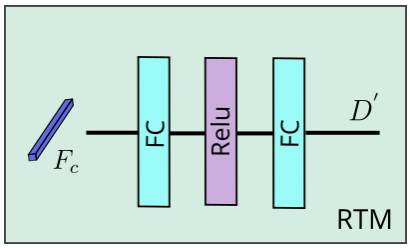

Fig. 3: The structure of RTM.

Token-adaptive Recalibration. To capture channel-wise interactions, F_c is projected by an MLP with ReLU activation, learning a weight vector $F_c^{'}$ which is used to re-weight the feature across the channels. $F_c^{'}$ is obtained by

$$F_c^{'} = \text{sigmoid}(\text{MLP}(F_c)). \tag{6}$$

Here, $F_c^{'} \in \mathbb{R}^d$ and sigmoid function is used to squeeze each element to a range of 0 and 1. We use a convolution layer to prepare F_p for the re-weighting and obtain $F_p^{'}$. After the recalibration, we add a skip connection [55] with F_p to derive the final feature map $F_f \in \mathbb{R}^{h_2 \times w_2 \times d}$, like

$$F_f = F_p + F_p^{'} \otimes F_c^{'}. \tag{7}$$

Through the TAM, the network can increase sensitivity to informative features which are important for downstream processing.

3.4 Regression-token Module (RTM)

Recall that the context token is used to collect global context over the whole image. It has a global overview of all image patches, through exchanging information with feature vector of each patch throughout all layers. Therefore, we adopt F_c to predict the overall count of people for the whole image. A two-layer MLP with ReLU activation is used to predict the total count $\hat{D}$, given by

$$\hat{D} = \text{MLP}(F_c). \tag{8}$$

The structure of RTM is shown in Fig. 3. We use L_1 loss to reduce the difference between $\hat{D}$ and ground-truth count, as follows,

$$\mathcal{L}_r(\hat{D}, \boldsymbol{D}) = |\hat{D} - \|\boldsymbol{D}\|_1|. \tag{9}$$

Note that we only use this module during training, the predicted count for an image during test is obtained by summing over the predicted density map $\boldsymbol{D}^{'}$ (§3.5), following other density-map based approaches [11, 29]. The benefits of RTM are two-fold. First, it forces to learn better context-token feature, which provides better information on the importance of each channel and enhance the final feature map F_f. Guiding the network to count the crowd through the context-token therefore encourages information exchange between context and patch tokens. In addition, it helps to learn better underlying feature representations and reduce over-fitting. This can be understood from a view of *auxiliary-task learning* [46, 56], which has shown to be effective in segmentation [38, 57].

3.5 Density Map Prediction and Loss Functions

To predict the density map $\boldsymbol{D}^{'}$, the feature map F_f is processed by a decoder containing two convolutional layers. We supervise the density map prediction using the distribution matching

loss introduced by [29]. To avoid dimension mismatch, the ground-truth $\boldsymbol{D}$ is resized to have the same size as $\boldsymbol{D}^{'}$, *i.e.*, $\boldsymbol{D} \in \mathbb{R}^{h_2 \times w_2}$. Specifically, the losses for learning the density map is a combination of counting loss, optimal transport loss [58] and variation loss, denoted as,

$$\mathcal{L}_d(\boldsymbol{D}^{'}, \boldsymbol{D}) = |\|\boldsymbol{D}^{'}\|_1 - \|\boldsymbol{D}\|_1| + \mathcal{L}_{OT}(\boldsymbol{D}^{'}, \boldsymbol{D}) + \mathcal{L}_{TV}(\boldsymbol{D}^{'}, \boldsymbol{D}), \tag{10}$$

where $\mathcal{L}_{OT}$ is the optimal transport loss, and $\mathcal{L}_{TV}$ is the total variation loss. The first term measures the difference of the total count between the predicted density map and the ground-truth binary mask. The second (optimal transport loss) and third terms (variation loss) are used to minimize the distribution difference between $\boldsymbol{D}^{'}$ and $\boldsymbol{D}$ by regarding the density map as a probability distribution. Please refer to [29] for more details. The total loss function for the proposed method is given by,

$$\mathcal{L} = \mathcal{L}_d(\boldsymbol{D}^{'}, \boldsymbol{D}) + \lambda_r \mathcal{L}_r(\hat{D}, \boldsymbol{D}), \tag{11}$$

where λ_r is the weight for the regression loss $\mathcal{L}_r$ from (9). The final predicted count for inference is the summation over the predicted density map, given by $\|\boldsymbol{D}^{'}\|_1$.

4 Experiments

We conduct extensive experiments on four benchmark crowd counting datasets [5, 7, 34, 35] to validate the effectiveness of the proposed approach. We begin this section by introducing our experimental setting, followed by comparisons with previous methods. Finally, we perform ablation studies to examine the effectiveness of different components of our model.

4.1 Experimental Setup

Implementation Details. The number of layers l in the main transformer is set to 14. We use the official T2T-ViT-14 model [43] pretrained on ImageNet [61] for initialization. For data augmentation, we adopt random cropping and random horizontal flipping in all experiments. We use the Adam optimizer [62], with learning rate and weight decay as 1e-5 and 1e-4, respectively. Following [37], we compute auxiliary losses at transformer layers T_5, T_8, and T_{11}, to provide intermediate supervision during training while only output from last layer is used for prediction. Our method is implemented in the PyTorch framework [63], and experiments are conducted on a single NVIDIA Tesla GPU. We will release our implementation for reproducibility.

Datasets. Experiments are conducted on four challenging datasets: ShanghaiTech [7], UCF-QNRF [5], JHU-CROWD++ [34] and NWPU [35]. ShanghaiTech contains 1,198 images with 330,165 annotations, and UCF-QNRF has 1,535 images with more than one million counts. JHU-CROWD++ and NWPU are two largest-scale and most challenging crowd counting benchmarks. JHU-CROWD++ consists of 4,822 images from diverse scenes with more than 1.5 million dot annotations, and NWPU contains 5,109 images with more than two million annotations. The results for the test set are obtained from the evaluation server.

Evaluation Metrics. Following previous works [3, 11, 29], we use mean average error (MAE) and mean square error (MSE) to evaluate the counting performance. For NWPU dataset, we also use mean normalized absolute error (NAE) as evaluation metric, following [29, 35].

4.2 Crowd Counting Results

Baseline. The baseline model is based on the same transformers, also adopting loss functions (11), without using TAM and regression-token module (RTM).

Quantitative Comparisons. For comparisons, we choose mainstream and popular methods. They can be divided into three groups. The VGG-based approaches include CSRNet [11], ic-CNN [12], CAN [3], PACNN [16], Wan *et al.* [19], PGCNet [21], BL [24], L2R [25], ASNet [26], LibraNet [27], Yang *et al.* [59], NoisyCC [28], DM-Count [29], MATT [60], and MBTTBF [64]. The ResNet-based methods include SFCN [15] and CG-DRCN [34]. Other CNN-based algorithms include Crowd CNN [6], MCNN [7], CMTL [8], Switch CNN [9], IG-CNN [10], CL-CNN [5], SANet [14], TEDnet [17], ANF [18], CFF [20].

The comparisons with other methods on various datasets are shown in Table 1, Table 2

Method	Dot	ShanghaiTech A		ShanghaiTech B		UCF-QNRF	
		MAE	MSE	MAE	MSE	MAE	MSE
Crowd CNN [6]	✓	181.8	277.7	32.0	49.8	-	-
MCNN [7]	✓	110.2	173.2	26.4	41.3	277	426
CMTL [8]	✓	101.3	152.4	20.0	31.1	252	514
Switch CNN [9]	✓	90.4	135.0	21.6	33.4	228	445
IG-CNN [10]	✓	72.5	118.2	13.6	21.1	-	-
CSRNet [11]	✓	68.2	115.0	10.6	16.0	-	-
ic-CNN [12]	✓	68.5	116.2	10.7	16.0	-	-
CL-CNN [5]	✓	-	-	-	-	132	191
SANet [14]	✓	67.0	104.5	8.4	13.6	-	-
CAN [3]	✓	62.3	100.0	7.8	12.2	107	183
SFCN [15]	✓	64.8	107.5	7.6	13.0	102	171
PACNN [16]	✓	62.4	102.0	7.6	11.8	-	-
TEDnet [17]	✓	64.2	109.1	8.2	12.8	113.0	188.0
ANF [18]	✓	63.9	99.4	8.3	13.2	110	174
Wan *et al.* [19]	✓	64.7	97.1	8.1	13.6	101	176
CFF [20]	✓	65.2	109.4	7.2	12.2	-	-
PGCNet [21]	✓	57.0	**86.0**	8.8	13.7	-	-
BL [24]	✓	62.8	101.8	7.7	12.7	88.7	154.8
L2R [25]	✓	73.6	112.0	13.7	21.4	124.0	196.0
ASNet [26]	✓	57.7	90.1	-	-	91.5	159.7
LibraNet [27]	✓	**55.9**	97.1	**7.3**	**11.3**	88.1	**143.7**
Yang *et al.* [59]	✗	104.6	145.2	12.3	21.2	-	-
NoisyCC [28]	✓	61.9	99.6	**7.4**	**11.3**	85.8	150.6
DM-Count [29]	✓	59.7	95.7	**7.4**	11.8	**85.6**	148.3
MATT [60]	✗	80.1	129.4	11.7	17.5	-	-
Baseline	✓	57.3	89.0	**7.4**	12.2	85.7	150.8
Ours	✓	**53.1**	**82.2**	**7.3**	**11.5**	**83.4**	**143.4**

Table 1: Comparison with state-of-the-art methods on ShanghaiTech A [7], ShanghaiTech B [7], and UCF-QNRF [5] datasets. The best and second best results are shown in **red** and **blue**, respectively.

Method	Publication	Dot	Val		Test	
			MAE	MSE	MAE	MSE
MCNN [7]	CVPR16	✓	160.6	377.7	188.9	483.4
CMTL [8]	AVSS17	✓	138.1	379.5	157.8	490.4
SANet [14]	ECCV18	✓	82.1	272.6	91.1	320.4
CSRNet [11]	CVPR18	✓	72.2	249.9	85.9	309.2
CAN [3]	CVPR19	✓	89.5	239.3	100.1	314.0
SFCN [15]	CVPR19	✓	62.9	247.5	77.5	297.6
BL [24]	ICCV19	✓	59.3	229.2	75.0	299.9
MBTTBF [64]	ICCV19	✓	73.8	256.8	81.8	299.1
CG-DRCN [34]	PAMI20	✓	57.6	244.4	71.0	278.6
Baseline	-	✓	**47.6**	**208.5**	**58.4**	**232.7**
Ours	-	✓	**46.5**	**198.6**	**54.8**	**208.5**

Table 2: Comparison with state-of-the-art methods on the JHU-CROWD++ dataset [34].

and Table 3. For all datasets, the proposed method performs favorably. It shows that our approach is stable across different datasets. The reported tables show that our baseline is already comparable to the state-of-the-art CNN-based methods. Notably, our full model outperforms the baseline model in almost all experiments, which validates the effectiveness of the proposed modules. In all cases, our method significantly outperforms DM-count [29], although both methods use the same decoder and loss functions to learn the density map.

Method	Publication	Dot	Val		Test		
			MAE	MSE	MAE	MSE	NAE
MCNN [7]	CVPR16	✓	218.5	700.6	232.5	714.6	1.063
CSRNet [11]	CVPR18	✓	104.8	433.4	121.3	387.8	0.604
CAN [3]	CVPR19	✓	93.5	489.9	106.3	386.5	0.295
SFCN [15]	CVPR19	✓	95.46	608.32	105.7	424.1	0.254
BL [24]	ICCV19	✓	93.64	470.38	105.4	454.2	0.203
KDMG [65]	PAMI20	✓	-	-	100.5	415.5	-
NoisyCC [28]	NeurIPS20	✓	-	-	96.9	534.2	-
DM-Count[29]	NeurIPS20	✓	70.5	357.6	88.4	388.6	**0.169**
Baseline	-	✓	**69.0**	**314.0**	**86.6**	**359.1**	0.172
Ours	-	✓	**53.0**	**170.3**	**82.0**	**366.9**	**0.164**

Table 3: Comparison with state-of-the-art crowd counting methods on the NWPU dataset [35].

For example, when compared with DM-count on ShanghaiTech A [7], our model reduces MAE from 59.7 to 53.1, and MSE from 95.7 to 82.2. This demonstrates the importance of global context features for the task of crowd counting.

On two largest-scale and most challenging benchmarks such as JHU-CROWD++ [34] and NWPU [35], our approach significantly outperforms the previous best results. More specifically, our method improves BL [24], the best method on JHU-CROWD++ test set, by reducing MAE from 75.0 to 54.8 and MSE from 299.9 to 208.5. Similarily on NWPU dataset, our method outperforms DM-count [29], the best method on the NWPU test set, by a margin of 6.4 and 21.7 on MAE and MSE, respectively. Note that the annotations for the NWPU test set are not publicly available and the corresponding results are obtained from the evaluation server.

Computing Time. Using an input image with size 256×256 and a Nvidia RTX 6000 GPU, the computing time of our method is 21.47 milliseconds while the time for DM-count [29] is 16.98 milliseconds. Note that DM-count is currently state-of-the-art CNN-based method and our method is based on transformer. Because of the use of transformer to establish global relation between features, our method consumes more time compared to CNN-based method. Developping more efficient crowd counting approach while using global information will be our future work.

Visualizations. Qualitative results of the predicted density maps are shown in Fig. 4. Our method generates sharper density maps and exhibits better localization ability, compared to DM-count [29].

4.3 Ablation Study

Following previous works [3, 11, 26, 27], we conduct ablation experiments on ShanghaiTech A [7], to show the contributions of the key components of our method.

TAM and RTM. Table 4a shows that the token-attention module and regression-token module provide complementary improvements over baseline. Specifically, by adding TAM to the baseline, we observe an improvement of 2.2 in MAE and of 2.8 in MSE. The best results are achieved by combining TAM with RTM, resulting in an improvement of 4.2 MAE and 6.8 in MSE over the baseline.

TAM vs. SE/CBAM. We also compare the proposed TAM block with a SE block [44] and a CBAM block [45]. The main difference between TAM and SE/CBAM is that the attention weight for TAM is obtained from context token, while SE/CBAM use feature itself to generate attention weight. As shown in Table 4a, TAM outperforms SE/CBAM, demonstrating that context token contains better information to recalibrate features along channels. The result for CBAM which uses both channel and spatial attention shows that additionally adding spatial attention does not help feature learning, since transformers are naturally equipped with spatial attention, as hypothesized in §3.3.

Sensitivity Analysis. Table 4b shows the results when varying λ_r controlling the contribution of $\mathcal{L}_r$ in Equation (11). We observe that our network is very robust to the choice of the λ_r parameter.

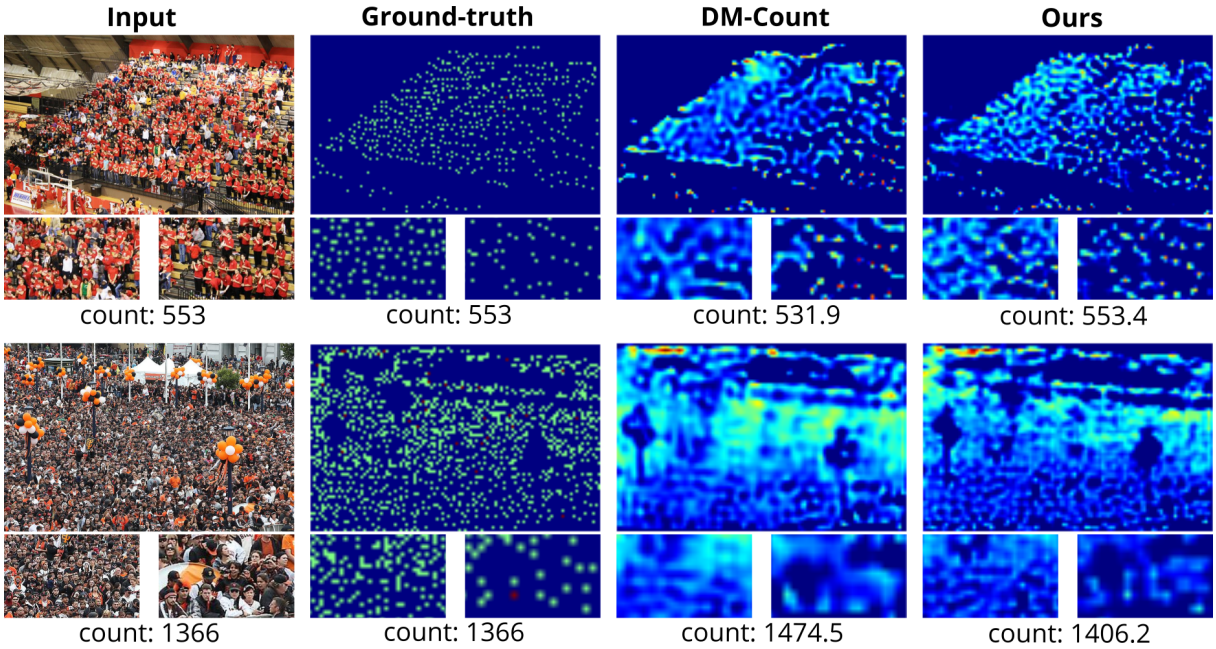

Fig. 4: **Density Map Visualization**. We compare the ground-truth density map, predicted density map from DM-count [29] and the proposed method. Our approach produces better density map for both dense and sparse regions, leading to more accurate count predictions.

Method	MAE	MSE
Baseline	57.3	89.0
Baseline+SE [44]	55.9	87.7
Baseline+CBAM [45]	56.2	90.0
Baseline+TAM	55.1	86.2
Baseline+SE+RTM [44]	54.6	84.2
Baseline+TAM+RTM	**53.1**	**82.2**

(a)

Method	λ_r	MAE	MSE
Ours	0.01	53.2	83.6
	0.1	**53.1**	**82.2**
	0.2	53.2	82.4
	0.5	53.3	82.4
	1.0	54.0	82.6

(b)

Table 4: Ablation study on **(a)** key components of our method and **(b)** λ_r on ShanghaiTech A.

4.4 Failure Cases and Limitation

Failure Cases. While our method achieves promising results on several datasets, there are cases where it does not perform well. We showed failure cases in Fig. 5. When the input images have low contrast or low quality, which do not frequently appear in the training set, our method does not predict the similar people count as the ground truth.

Limitation. This paper aims at exploiting the effect of global context in crowd counting. Although we have achieved this goal by demonstrating the effectiveness of the proposed context extraction techniques, we do not explore how to incorporate our techniques into existing state-of-the-art counting methods [27–29, 60] for performance boosting. We leave this as the future work as this is totally about engineering. Moreover, we find that the improvement of our context techniques on large datasets [34, 35] is much more significant than that on small datasets [5, 7]. This may be a platitude that deep learning needs large-scale data to evaluate its real performance. Hence, we suggest researchers paying more attention to recent large-scale datasets [34, 35] in the future.

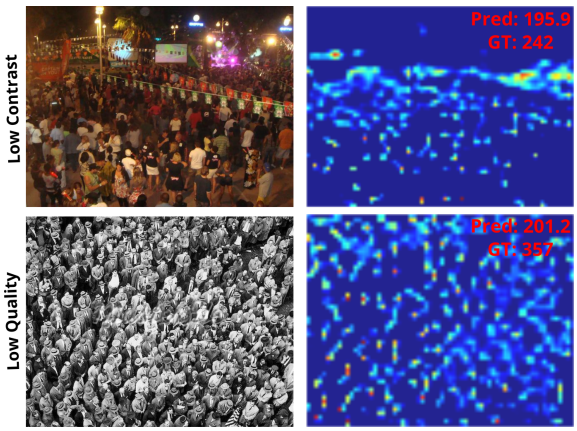

Fig. 5: Failure Cases. The failure cases are caused by the *low contrast* or *low quality* of the input images.

5 Conclusion

In this paper, we study the value of global context information in crowd counting with point supervision. We build a strong baseline using transformers to encode features with global receptive fields. Based on that, we proposed two novel modules: token-attention module and regression-token module. Extensive experiments are conducted to validate the effectiveness of the proposed techniques. Our context techniques achieve significant improvement on ShanghaiTech [7], UCF-QNRF [5], JHU-CROWD++ [34], and NWPU [35] datasets. Therefore, we conclude that facilitating the representation of global context significantly benefits crowd counting.

References

[1] W. Ge and R. T. Collins, "Marked point processes for crowd counting," in *IEEE CVPR*, 2009, pp. 2913–2920.

[2] T. Zhao and R. Nevatia, "Bayesian human segmentation in crowded situations," in *IEEE CVPR*, vol. 2, 2003, pp. 452–459.

[3] W. Liu, M. Salzmann, and P. Fua, "Context-aware crowd counting," in *IEEE CVPR*, 2019, pp. 5099–5108.

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

[5] H. Idrees, M. Tayyab, K. Athrey, D. Zhang, S. Al-Maadeed, N. Rajpoot, and M. Shah, "Composition loss for counting, density map estimation and localization in dense crowds," in *ECCV*, 2018, pp. 532–546.

[6] C. Zhang, H. Li, X. Wang, and X. Yang, "Cross-scene crowd counting via deep convolutional neural networks," in *IEEE CVPR*, 2015, pp. 833–841.

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

[8] V. A. Sindagi and V. M. Patel, "CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting," in *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1–6.

[9] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in *IEEE CVPR*. IEEE, 2017, pp. 4031–4039.

[10] D. B. Sam, N. N. Sajjan, R. V. Babu, and M. Srinivasan, "Divide and grow: Capturing huge diversity in crowd images with incrementally growing CNN," in *IEEE CVPR*, 2018, pp. 3618–3626.

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

[12] V. Ranjan, H. Le, and M. Hoai, "Iterative crowd counting," in *ECCV*, 2018, pp. 270–285.

[13] Z. Shi, L. Zhang, Y. Liu, X. Cao, Y. Ye, M.-M. Cheng, and G. Zheng, "Crowd counting with deep negative correlation learning," in *IEEE CVPR*, 2018, pp. 5382–5390.

[14] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *ECCV*, 2018, pp. 734–750.

[15] Q. Wang, J. Gao, W. Lin, and Y. Yuan, "Learning from synthetic data for crowd counting in the wild," in *IEEE CVPR*, 2019, pp. 8198–8207.

[16] M. Shi, Z. Yang, C. Xu, and Q. Chen, "Revisiting perspective information for efficient crowd counting," in *IEEE CVPR*, 2019, pp. 7279–7288.

[17] X. Jiang, Z. Xiao, B. Zhang, X. Zhen, X. Cao, D. Doermann, and L. Shao, "Crowd counting and density estimation by trellis encoder-decoder networks," in *IEEE CVPR*, 2019, pp. 6133–6142.

[18] A. Zhang, L. Yue, J. Shen, F. Zhu, X. Zhen, X. Cao, and L. Shao, "Attentional neural fields for crowd counting," in *IEEE ICCV*, 2019, pp. 5714–5723.

[19] J. Wan and A. Chan, "Adaptive density map generation for crowd counting," in *IEEE ICCV*, 2019, pp. 1130–1139.

[20] Z. Shi, P. Mettes, and C. G. Snoek, "Counting with focus for free," in *IEEE ICCV*, 2019, pp. 4200–4209.

[21] Z. Yan, Y. Yuan, W. Zuo, X. Tan, Y. Wang, S. Wen, and E. Ding, "Perspective-guided convolution networks for crowd counting," in *IEEE ICCV*, 2019, pp. 952–961.

[22] H. Xiong, H. Lu, C. Liu, L. Liu, Z. Cao, and C. Shen, "From open set to closed set: Counting objects by spatial divide-and-conquer," in *IEEE ICCV*, 2019, pp. 8362–8371.

[23] L. Liu, Z. Qiu, G. Li, S. Liu, W. Ouyang, and L. Lin, "Crowd counting with deep structured scale integration network," in *IEEE ICCV*, 2019, pp. 1774–1783.

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

[25] X. Liu, J. Van De Weijer, and A. D. Bagdanov, "Exploiting unlabeled data in CNNs by self-supervised learning to rank," *IEEE TPAMI*, vol. 41, no. 8, pp. 1862–1878, 2019.

[26] X. Jiang, L. Zhang, M. Xu, T. Zhang, P. Lv, B. Zhou, X. Yang, and Y. Pang, "Attention scaling for crowd counting," in *IEEE CVPR*, 2020, pp. 4706–4715.

[27] L. Liu, H. Lu, H. Zou, H. Xiong, Z. Cao, and C. Shen, "Weighing counts: Sequential crowd counting by reinforcement learning," in *ECCV*, 2020, pp. 164–181.

[28] J. Wan and A. Chan, "Modeling noisy annotations for crowd counting," in *NeurIPS*, 2020, pp. 3386–3396.

[29] B. Wang, H. Liu, D. Samaras, and M. Hoai, "Distribution matching for crowd counting," in *NeurIPS*, 2020.

[30] Z. Ma, X. Wei, X. Hong, and Y. Gong, "Learning scales from points: A scale-aware probabilistic model for crowd counting," in *ACM Multimedia*, 2020, pp. 220–228.

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

[32] K. Kopaczewski, M. Szczodrak, A. Czyzewski, and H. Krawczyk, "A method for counting people attending large public events," *Multimedia Tools and Applications*, vol. 74, no. 12, pp. 4289–4301, 2015.

[33] J. He, X. Wu, J. Yang, and W. Hu, "Cpspnet: Crowd counting via semantic segmentation framework," in *2020 IEEE 32nd International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2020, pp. 1104–1110.

[34] V. Sindagi, R. Yasarla, and V. M. Patel, "JHU-CROWD++: Large-scale crowd counting dataset and a benchmark method," *IEEE TPAMI*, 2020.

[35] Q. Wang, J. Gao, W. Lin, and X. Li, "NWPU-Crowd: A large-scale benchmark for crowd counting and localization," *IEEE TPAMI*, 2020.

[36] J. Watson, O. Mac Aodha, D. Turmukhambetov, G. J. Brostow, and M. Firman, "Learning stereo from single images," in *ECCV*, 2020, pp. 722–740.

[37] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *IEEE CVPR*, 2021.

[38] G. Sun, W. Wang, J. Dai, and L. Van Gool, "Mining cross-image semantics for weakly supervised semantic segmentation," in *ECCV*, 2020, pp. 347–365.

[39] G. Yang, H. Tang, M. Ding, N. Sebe, and E. Ricci, "Transformers solve the limited receptive field for monocular depth prediction," *arXiv preprint arXiv:2103.12091*, 2021.

[40] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *NeurIPS*, 2021.

[41] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 6000–6010.

[42] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[43] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, F. E. Tay, J. Feng, and S. Yan, "Tokens-to-token ViT: Training vision transformers from scratch on imagenet," *arXiv preprint arXiv:2101.11986*, 2021.

[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *IEEE CVPR*, 2018, pp. 7132–7141.

[45] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *ECCV*, 2018, pp. 3–19.

[46] S. Gidaris, P. Singh, and N. Komodakis, "Unsupervised representation learning by predicting image rotations," in *ICLR*, 2018.

[47] K. Chen, S. Gong, T. Xiang, and C. Change Loy, "Cumulative attribute space for age and crowd density estimation," in *IEEE CVPR*, 2013, pp. 2467–2474.

[48] C. Wang, H. Zhang, L. Yang, S. Liu, and X. Cao, "Deep people counting in extremely dense crowds," in *ACM Multimedia*, 2015, pp. 1299–1302.

[49] M. Li, Z. Zhang, K. Huang, and T. Tan, "Estimating the number of people in crowded scenes by mid based foreground segmentation and head-shoulder detection," in *IEEE ICPR*, 2008, pp. 1–4.

[50] C. Wang, Q. Song, B. Zhang, Y. Wang, Y. Tai, X. Hu, C. Wang, J. Li, J. Ma, and Y. Wu, "Uniformity in heterogeneity: Diving deep into count interval partition for crowd counting," in *IEEE ICCV*, 2021, pp. 3234–3242.

[51] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Wu, "Rethinking counting and localization in crowds: A purely point-based framework," in *IEEE ICCV*, 2021, pp. 3365–3374.

[52] X. Liu, G. Li, Z. Han, W. Zhang, Y. Yang, Q. Huang, and N. Sebe, "Exploiting sample correlation for crowd counting with multi-expert network," in *IEEE ICCV*, 2021, pp. 3215–3224.

[53] D. Liang, X. Chen, W. Xu, Y. Zhou, and X. Bai, "TransCrowd: Weakly-supervised crowd counting with transformer," *arXiv preprint arXiv:2104.09116*, 2021.

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

[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016, pp. 770–778.

[56] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," in *ICML*. PMLR, 2017, pp. 2642–2651.

[57] Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *ECCV*, 2020, pp. 282–298.

[58] C. Villani, *Optimal transport: old and new*. Springer Science & Business Media, 2008, vol. 338.

[59] Y. Yang, G. Li, Z. Wu, L. Su, Q. Huang, and N. Sebe, "Weakly-supervised crowd counting learns from sorting rather than locations," in *ECCV*, 2020.

[60] Y. Lei, Y. Liu, P. Zhang, and L. Liu, "Towards using count-level weak supervision for crowd counting," *Pattern Recognition*, vol. 109, p. 107616, 2021.

[61] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "ImageNet large scale visual recognition challenge," *IJCV*, vol. 115, no. 3, pp. 211–252, 2015.

[62] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[63] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8026–8037.

[64] V. A. Sindagi and V. M. Patel, "Multi-level bottom-top and top-bottom feature fusion for crowd counting," in *IEEE ICCV*, 2019, pp. 1002–1012.

[65] J. Wan, Q. Wang, and A. B. Chan, "Kernel-based density map generation for dense object counting," *IEEE TPAMI*, 2020.