Hypergraph Association Weakly Supervised Crowd Counting

BO LI, YONG ZHANG, CHENGYANG ZHANG, XINGLIN PIAO, and BAOCAI YIN, Beijing University of Technology

Weakly supervised crowd counting involves the regression of the number of individuals present in an image, using only the total number as the label. However, this task is plagued by two primary challenges: the large variation of head size and uneven distribution of crowd density. To address these issues, we propose a novel Hypergraph Association Crowd Counting (HACC) framework. Our approach consists of a new multi-scale dilated pyramid module that can efficiently handle the large variation of head size. Further, we propose a novel hypergraph association module to solve the problem of uneven distribution of crowd density by encoding higher-order associations among features, which opens a new direction to solve this problem. Experimental results on multiple datasets demonstrate that our HACC model achieves new state-of-the-art results.

CCS Concepts: • **Computing methodologies** → *Scene understanding;*

Additional Key Words and Phrases: Crowd counting, hypergraph neural network, uneven distribution of crowd density, hypergraph association

ACM Reference format:
Bo Li, Yong Zhang, Chengyang Zhang, Xinglin Piao, and Baocai Yin. 2023. Hypergraph Association Weakly Supervised Crowd Counting. *ACM Trans. Multimedia Comput. Commun. Appl.* 19, 6, Article 195 (July 2023), 20 pages.
https://doi.org/10.1145/3594670

1 INTRODUCTION

Given the pressing needs of social security, crowd counting has emerged as a topic of increasing interest to both academia and industry. With frequent occurrences of crowd gathering in various public places such as station halls, terminal buildings, cinemas, and shopping malls, crowd counting plays a critical role in ensuring safety, managing traffic, and planning spatial arrangements in these settings. Moreover, the crowd counting model can be readily extended to other domains such as vehicle counting [4, 42], cell counting [15, 23], and crowd video analysis [21].

This research project was partially supported by the National Key R&D Program of China (2021ZD0111902), the National Natural Science Foundation of China (62072015, U21B2038, U19B2039, and 61902053), and the Beijing Natural Science Foundation (4222021).

Authors' address: B. Li, Y. Zhang (corresponding author), C. Zhang, X. Piao, and B. Yin, Beijing Key Laboratory of Multimedia and Intelligent Software Technology, Beijing Artificial Intelligence Institute, Faculty of Information Technology, Beijing University of Technology, Beijing, China, 100124; emails: bo_li@emails.bjut.edu.cn, zhangyong2010@bjut.edu.cn, Cy_Zhang@emails.bjut.edu.cn, {piaoxl@bjut.edu.cn, ybc}@bjut.edu.cn.

1551-6857/2023/07-ART195 $15.00
https://doi.org/10.1145/3594670

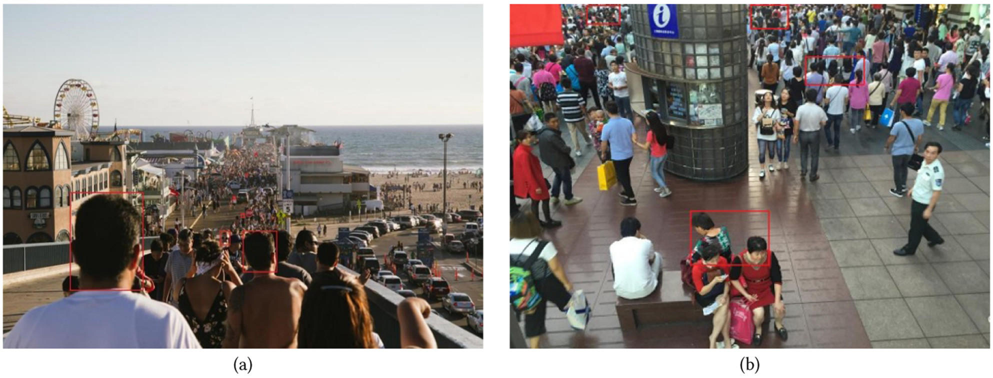


Fig. 1. The main problems in crowd counting: the large variation of head size; (a) and the uneven distribution of crowd density (b). Best viewed in red boxes.

The task of crowd counting faces two primary challenges: the large variation of head size and the uneven distribution of crowd density, as shown in Figure 1. The former is primarily due to the phenomenon of near large and far small, whereas the latter is attributed to the fact that crowds tend to aggregate. To address these issues, researchers have developed a variety of effective methods that can be broadly categorized into three groups: detection-based [30, 47], regression-based [25, 67], and **Convolutional Neural Network (CNN)**-based methods [20, 37]. Among these, CNN-based methods have been widely studied and have achieved significant success in recent years. The current CNN-based crowd counting approaches can be divided into two branches: strongly supervised and weakly supervised. Strongly supervised methods require detailed point-level annotation information about the head position of each individual, whereas weakly supervised methods only require total count information.

Strongly supervised crowd counting methods utilize density maps [32] as prior information, transforming the task from an image-to-number relationship to an image-to-image mapping problem. To address the issue of the large variation of head size, researchers typically use multiple parallel network branches to provide features with different receptive fields [5, 59, 64, 81]. However, Li et al. [36] have demonstrated that the features learned by these multiple branches are often similar, as seen in the case of **Multi-Column Convolutional Neural Network (MCNN)** [81]. To overcome this limitation, some researchers have utilized Inception [63] and its variants [6, 76, 78] to extract multi-scale information, which have fewer parameters and greater aggregation power. As an example, the MSCNN [78] model employs four parallel convolutional branches with kernel sizes of 3, 5, 7, and 9 to extract features. However, such methods have two primary drawbacks. First, traditional convolutional operations may impede dense prediction tasks and result in the loss of spatial information within features [8, 9, 44]. Second, in general, using larger convolution kernels increases computational cost and requires more parameters and computational workload.

To address the issue of the uneven distribution of crowd density, some crowd counting methods adopt patch-based processing or attention mechanisms to focus on dense regions of the image [3, 20, 53, 54, 83, 84]. However, these methods require point-level annotations of head positions, which can be costly and unnecessary for evaluation in the testing stage [37]. In contrast, weakly supervised crowd counting methods only require total head count as supervision information, making them more suitable for real-world scenarios. Recently, Liang et al. [37] introduced transformers to weakly supervised crowd counting and achieved substantial performance improvements.

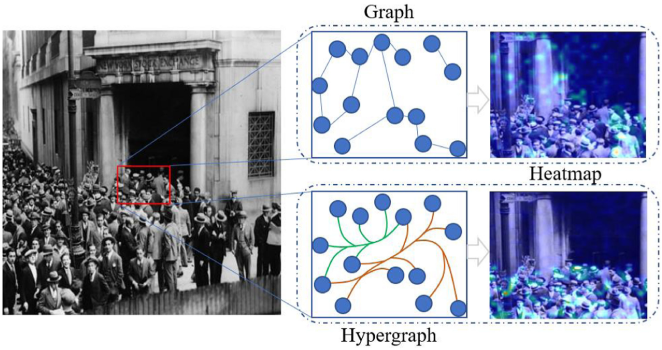


Fig. 2. The disparities in construction methods and heatmaps between graphs and hypergraphs. In certain crowd counting scenarios, the appearance of local crowds in clusters with similar characteristics is a common occurrence. To achieve a holistic representation of these local clusters in an image, a methodology that can concurrently aggregate multiple node features is necessary. Nevertheless, this goal poses a challenge for traditional graph techniques, as they only utilize pairwise connections between data nodes. Conversely, hypergraph techniques provide degree-free hyperedges, enabling the simultaneous aggregation and representation of multiple node features, which aligns with our need for representing local clusters. Consequently, hypergraphs provide a more reasonable heatmap, which proves to be a difficult task to accomplish with graphs.

Furthermore, CrowdFormer [55] enhanced counting performance by fusing features from various levels in the transformer backbone to generate multi-scale features with contextual information. However, there is still a performance disparity between weakly and strongly supervised methods, and none of these methods have addressed the issue of uneven distribution of crowd density.

Although the CNN-based methods discussed previously have yielded promising crowd counting results, there remain two challenges. First, the uniform transformation in CNNs struggles to model the relationships among features. Second, the local receptive domain, which limits the feature aggregation capability, cannot accommodate complex topologies. To address these issues, researchers have incorporated graphs into CNNs, giving rise to **Graph Convolutional Networks (GCNs)** [2, 13, 29]. GCNs can better represent input data by encoding graph structures and aggregating relevant information from a global perspective. This approach overcomes the limitations of feature aggregation to the local receptive domain and enhances joint feature characterization. However, classical GCNs only utilize pairwise connections between data nodes, which cannot capture the complex relational representations required in practical scenarios. In the context of crowd counting, local crowds frequently cluster together and exhibit similar characteristics. To represent these local clusters holistically, a methodology that can simultaneously aggregate multiple node features is required. However, traditional CNNs and GCNs struggle to achieve this objective. The **HyperGraph Neural Network (HGNN)** [18], which introduces degree-free hyperedges, can aggregate and represent multiple node features simultaneously. This feature aligns with the need to represent local clusters in crowd counting scenarios, as shown in Figure 2. We have therefore introduced the HGNN method and applied it to crowd counting. However, most existing HGNN-based methods [18, 27, 51] use a simple Euclidean distance between nodes to construct hyperedges, which cannot accommodate complex semantic associations among features.

This article proposes a new weakly supervised **Hypergraph Association Crowd Counting (HACC)** framework, which consists of a **Multi-scale Dilated Pyramid (MDP)** module and a new **Hypergraph Association (HA)** module. To effectively capture multi-level information on the feature map, an MDP module is designed with varying dilated rates, which preserves the spatial structure information of features while minimizing computational costs. Additionally, to address the issue of uneven crowd density, we introduce HGNN into the field of crowd counting, which models the relationship among features. Moreover, we propose a new HA module, which builds hypergraphs based on Euclidean distance and learnable similarity association. Our experiments show that the module can achieve scene understanding.

The main contributions of this article are summarized as follows:

- To effectively address the problem of the large variation of head size, we design an MDP module to capture multi-scale features while retaining the internal structure of the features to the greatest extent.
- To overcome the challenge of uneven distribution of crowd density, we introduce HGNNs to the crowd counting domain for the first time. Furthermore, we propose a novel HA module that leverages both Euclidean distances and learnable similarity associations to better capture the correlations among individuals.
- We conduct extensive experiments on various benchmark datasets, including JHU-CROWD++, Shanghai Tech A/B, UCF-QNRF, and UCF_CC_50. The experimental results demonstrate that our proposed HACC model outperforms state-of-the-art methods.

The article is organized as follows. We discuss the related works in Section 2, and the proposed HACC model is discussed in detail in Section 3. Section 4 shows the experimental results and analysis, Section 5 develops a discussion of the hypergraph module, and finally, Section 6 presents our conclusion and outlook.

2 RELATED WORKS

In this section, we briefly discuss the research status of strongly supervised, weakly supervised, and graph-based crowd counting works.

2.1 Strongly Supervised Crowd Counting

In addressing the problem of uneven distribution of crowd density, two strongly supervised approaches are commonly employed: patch-based and attention-based methods. Patch-based methods partition the image into multiple patches and process each patch using specialized techniques [3, 53, 54]. These methods utilize multi-branch networks with varying receptive fields, and a classification network determines which branch network processes each patch. In this way, the processing methods of patches with different densities are diverse. However, these methods are ineffective at handling uneven density distributions within each patch and do not address the issue fundamentally. Alternatively, attention-based methods employ the attention mechanism for densely populated areas [20, 83, 84]. These methods typically consist of two modules: channel attention, which extracts the most significant features from the channels, making the model more resilient to noisy backgrounds, and spatial attention, which enables the model to focus on the crowded areas of the image.

The efficacy of crowd counting techniques that rely heavily on comprehensive point-level supervision is notable. However, in practical settings, such supervision may not always be readily available, thus impeding the broader applicability of these methods. Consequently, researchers are endeavoring to transition to count-level annotation as an alternative approach.

2.2 Weakly Supervised Crowd Counting

Lei et al. [31] have introduced MATT (Multiple Auxiliary Tasks Training), an effective yet straightforward training strategy to impose constraints on the generated density maps. In the absence of point-level annotations, Yang et al. [77] have presented a soft label classification network and a counting network that classify images based on the number of individuals present in each image. More recently, Liang et al. [37] have incorporated transformers into weakly supervised crowd counting by leveraging ViT (the Vision Transformer [14]) as the backbone network to capture global semantic information, whereas a simple regression head is employed to predict crowd numbers. Tian et al. [65] have proposed the CCTrans method, which leverages a pyramid vision transformer to extract multi-level features and a simple multi-scale atrous convolution regression head to estimate the crowd number. Similarly, Savner and Kanhangad [55] have introduced the CrowdFormer method, which employs a pyramid structured vision transformer to extract multi-scale features with global context and combines them to estimate the crowd number.

Despite the cost reduction achieved by count-level annotation-based weakly supervised crowd counting methods, their performance still falls short of achieving results comparable to those of strongly supervised counting methods.

2.3 Graph-Based Crowd Counting

Luo et al. [50] have pioneered the application of HyGNN (Hybrid GNN) to mine the relationship between crowd localization and crowd counting within a single image. This approach views the feature maps of varying scales as nodes and identifies two types of relations as edges. By continuously aggregating and updating node features, HyGNN can capture richer associations and achieve stronger representation capabilities. Chen et al. [10] have devised a region relationship sensing module based on GNNs to identify and explore relationships among regions with different densities. Li et al. [34] have introduced GGRNet (the Graph-based Global Reasoning Network), which leverages GGRU (the Graph-based Global Reasoning Unit) to reason about context information from the features obtained through VGG-16. Furthermore, Zhai et al. [79] have proposed a graph-based multi-view learning model named *CoCo-GCN* (the Co-Communication Graph Convolutional Network) for multi-view crowd counting. This approach jointly investigates contextual dependencies and captures complementary relationships across different views.

The preceding works have demonstrated effective performance in crowd counting through the utilization of robust node feature representation ability of the **Graph Neural Network (GNN)**. However, it is important to note that the representation capability of these models is restricted due to the inherent limitation of the GNN, which only allows for pairwise connections between nodes. As a result, it is not possible to accurately capture higher-order association relationships, as illustrated in Figure 2.

3 OUR METHOD

The overview of our HACC method is shown in Figure 3. It mainly contains three modules: the transformer backbone, MDP module, and HA module. First, an image is input into the backbone network based on the Swin transformer [48] and **Feature Pyramid Network (FPN)** [41] to extract feature information at various stages. By concatenating features from different stages, we obtain the original feature map with global semantic information and local fine-grained information. Second, the MDP module is designed to capture information at different levels of the same feature map, using different dilated rates but with the same depth. Finally, a 1×1 convolution is applied to adjust the channels of the structural feature map, which is then fed into the HA module to model the correlation among features.

3.1 Transformer Backbone

The transformer backbone is built based on the Swin transformer [48] and FPN [41] structure. The Swin transformer is a hierarchical vision backbone that has found widespread use in various downstream vision tasks. It consists of four stages that produce output feature maps with decreasing

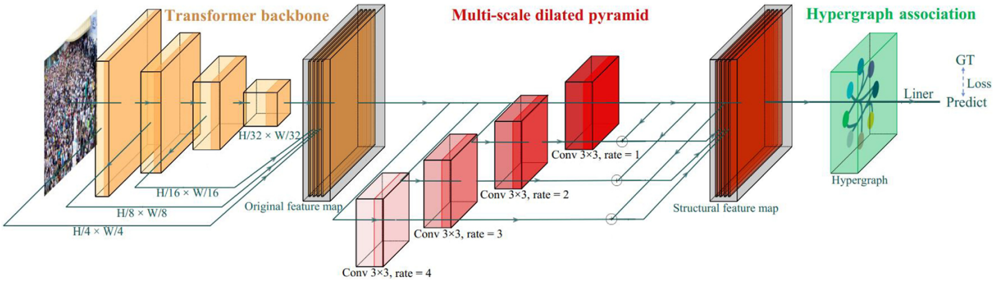


Fig. 3. The framework of HACC. It comprises three primary modules: a transformer backbone, an MDP module, and a HA module. Initially, the image features are fed through the transformer backbone, and the resulting features of the four stages are concatenated to create an original feature map that contains both global and local information. Subsequently, the MDP module is utilized to capture features at various levels and preserve the spatial information of these features. The HA module is then employed to model the association among these features, thereby facilitating scene understanding. Finally, a straightforward linear mapping is applied to the feature map to generate the predicted number of individuals.

resolution. Its primary innovation is the application of the sliding window concept to the transformer. This involves using non-overlapping local windows and overlapping cross windows to limit the most expensive attentional computations in the transformer to a single window. This approach introduces locality to the convolution operation while also significantly reducing the amount of computation cost. FPN, however, aims to construct feature maps of different scales by utilizing a top-down and laterally connected network structure, which incorporates global image information into low-level features. To achieve this, we input the feature maps produced by the four stages of transformer into the corresponding level of FPN for information fusion. Since the FPN structure increases the model's parameters, we use group convolution (with four groups) to reduce the number of parameters. Finally, we merge the features output by the FPN structure at different levels to obtain a feature representation that contains both global and fine-grained information. We then input this representation into the MDP module to extract a multi-level feature structured representation.

3.2 MDP Module

Upon obtaining the feature map outputted by the transformer backbone, both global and local fine-grained features are initially acquired. To capture features at various levels and maintain the spatial information of features to some extent, the features from different stages are stacked together. Moreover, in contrast to traditional large kernel convolution, dilated convolution can increase the receptive field while preserving the spatial structure information of the features, while also decreasing computational costs. Therefore, the MDP module is devised, utilizing different dilated rates to acquire varying levels of information from the feature map.

To be specific, the feature map's channel number is first adjusted, after which it is inputted into the multi-scale dilated convolution. Drawing inspiration from previous works [8, 9, 35, 44], we design four dilated convolution branches with varying receptive fields to capture objects of different scales. In contrast to object detection scenarios, crowd counting scenarios often involve a large number of small-sized heads, which we believe necessitates prioritizing fine-grained features for accuracy. Therefore, we set the dilated rates of the four branches to be smaller (1, 2, 3, and 4), allowing the network to capture features of different scales, mitigate the issue of large variations of head size, and avoid the grid effect [17]. As shown in Figure 4, the MDP module leverages four branches with different dilated rates to capture objects of varying scales in the original feature map.

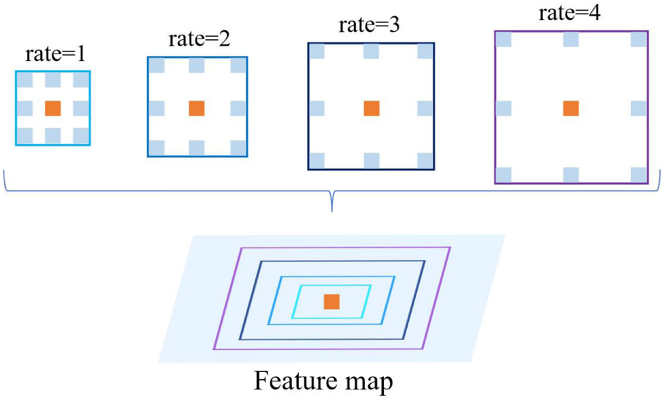


Fig. 4. Hierarchical correspondence representation between the MDP module and feature map. The module employs multiple branches with different dilated rates of 1, 2, 3, and 4 to capture heads of various scales within the feature map. By utilizing this approach, the spatial structure and hierarchy information of the features are preserved.

Additionally, given the small and densely packed nature of heads in crowd counting scenarios, we emphasize the importance of low-level, fine-grained features. Hence, we incorporate the output of the branch with a dilated rate of 1 into the other three branches, further augmenting the fine-grained information.

After obtaining the features from the aforementioned branches, we merge them and input them into the HA module. Notably, the MDP module successfully preserves the data structure and spatial hierarchy information of the features, which is advantageous for the subsequent HA module to establish higher-order semantic associations among features, facilitating scene understanding.

3.3 HA Module

Further, our objective with the HACC model is to effectively identify key areas in complex counting scenes, develop scene understanding capabilities, and address issues with uneven distribution of crowd density. To this end, we introduce a novel HA module that jointly describes local individuals with similar features in images by leveraging both Euclidean distance and learnable similarity correlations among features.

Given a feature map as $F \in \mathbb{R}^{C\times H\times W}$, to generate a hypergraph, we first change the feature dimension to $F \in \mathbb{R}^{N\times N}$, and the grid of $N \times 1$ is treated as a node $\mathcal{V}_i$ with feather f_i. Typically, the association between node features in an image is stronger when the Euclidean distance between them is smaller. Therefore, the similarity between two nodes $\mathcal{V}_i$ and $\mathcal{V}_j$ is initially measured using the Euclidean distance, which is computed as follows:

$$M_{dis} = \sqrt{\sum^{|N|}(f_i - f_j)^2}, \tag{1}$$

where N denotes the number of nodes in the feature map, and f_i and f_j refer to the feature vectors corresponding to $\mathcal{V}_i$ and $\mathcal{V}_j$, respectively. However, relying solely on the Euclidean distance metric may not be sufficient to distinguish human heads from similarly backgrounds in some crowd counting scenarios. To address this limitation, we propose a weighted cosine similarity approach. Cosine similarity measures the angle between feature vectors rather than their absolute sizes, making it more appropriate for high-dimensional feature spaces. Unlike Euclidean distance, cosine similarity is more sensitive to similarities between feature vectors and is thus more effective at mitigating confusion between human heads and their surrounding backgrounds. The cosine similarity can be calculated using the following equation:

$$Cos(\Theta) = \frac{f_i f_j^T}{||f_i|| \cdot ||f_j||}. \tag{2}$$

Building on previous works [22, 72, 74], our aim is to enable the model to learn the similarity between features adaptively, allowing it to better capture the relationship between human heads and similarly backgrounds. To achieve this, we introduce learnable parameters into the cosine similarity equation. The resulting learnable cosine similarity M_{sim} between nodes $\mathcal{V}_i$ and $\mathcal{V}_j$ is represented as follows:

$$Sim(f_i, f_j) = \frac{(f_i W_i)(f_j W_j)^T}{|f_i W_i|_2 \cdot |f_j W_j|_2}, \tag{3}$$

where W_i and W_j are the learnable weights, and $|\ |_2$ denotes the L_2 norm. In summary, we combine the Euclidean distance M_{dis} and the learnable cosine similarity M_{sim} to measure the similarity between node features. The association score between node $\mathcal{V}_i$ and $\mathcal{V}_j$ is denoted by

$$M_{ij} = \alpha \times M_{dis} + \beta \times M_{sim}. \tag{4}$$

Furthermore, it is worth considering that the measurement methods for node features used to calculate the similarity metrics, M_{dis} and M_{sim}, may differ significantly. As a result, the two metrics could exhibit a large discrepancy in their values, leading to the potential loss of the weighted effect of one of them. To mitigate this issue, we propose to normalize M_{dis} and M_{sim} separately using the softmax function.

ALGORITHM 1: Hypergraph Construction

```
Input: Embedding $X$
Function: Construct hyperedge based on Euclidean distance and association relations.
1: Generate $M_{sim}$ according to Equation (3)
2: for m in $X$ do
3:     $M_{tmp}$ = Eu_dis(m)
4:     $M_{dis}$.append($M_{tmp}$)
5: end for
6: $M = \alpha \times M_{dis} + \beta \times M_{sim}$
7: for i in range(len($M$)) do
8:     $H_{tmp}$ = Construct_H_with_KNN($M$[i])
9:     $H$.append($H_{tmp}$)
10: end for
Output: incidence matrix $H$
```

Subsequently, the nodes in M_{ij} are selected as centroids in turn and form a hyperedge with each of the nine nodes with the maximum correlation score. By the preceding operation, the matrix H is obtained, as shown in Algorithm 1, and then input it into a four-layer hypergraph convolution. A complete hypergraph convolution layer is obtained by adding a non-linear activation function to the hypergraph convolution operation, which can be expressed as

$$X^{l+} = \sigma\left(D_v^{-1/2} H W D_e^{-1} H^\top D_v^{-1/2} X^{(l)} \Theta^{(l)}\right), \tag{5}$$

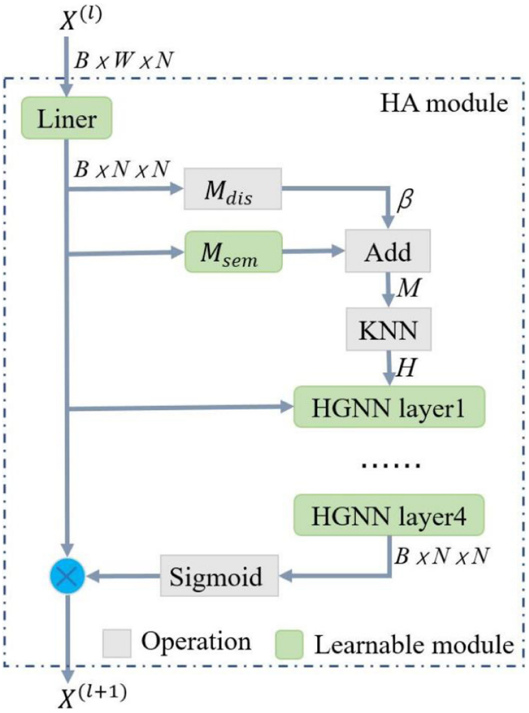


Fig. 5. The processing flow of the HA module.

where $X^{(l+)}$ is the output of the l-th layer and σ is the ReLU function used for nonlinear activation. D_v and D_e denote the diagonal matrix of vertex degrees and edge degrees, with each vertex degree defined as $d(v) = \sum_{e \in \mathcal{E}} \omega(e)h(v,e)$ and each edge degree defined as $\delta(e) = \sum_{v \in \mathcal{V}} h(v,e)$. The role of D_v and D_e can be simply summarized as normalizing incidence matrix H. Both W and Θ are learnable parameters. Finally, considering that the optimized features X^{l+1} are too smooth, we adopt them as attention weights to optimize the features:

$$X^{l+1} = Sigmoid(X^{l+}) \cdot X^{l}. \tag{6}$$

More specifically, the processing flow of the HA module is shown in Figure 5. First, for the structural feature map $X^{(l)}$ obtained from the MDP module, we utilize a liner layer to change its dimension to $\mathbb{R}^{B \times N \times N}$. Second, the features are sent into the learnable similarity correlations matrix M_{sim} and distance matrix M_{dis} to generate the final matrix M based on Equation (4). Then we can obtain the incidence matrix H using the nearest neighbor algorithm based on M. Third, incidence matrix H and the features X^{l} are sent to a four-layer hypergraph convolution to enhance the representation ability of features. Finally, the refined feature map $X^{(l+1)}$ is output according to Equation (6).

4 EXPERIMENT RESULTS AND ANALYSIS

4.1 Datasets and Experimental Details

The datasets in the field of weakly supervised crowd counting mainly include JHU-CROWD++ [57], Shanghai Tech A/B [81], UCF-QNRF [26], and UCF_CC_50 [25]. Next, we briefly describe the characteristics of each dataset.

The *JHU-CROWD++* [57] dataset is an extension based on JHU-Crowd [61] and contains a total of 4,372 images and 1.51 million instances with an average resolution of 910 × 1430. The dataset contains some scenes under severe weather and lighting conditions, such as snow, rain, and haze, and provides rich crowd head location annotations.

The *Shanghai Tech* [81] dataset is composed of two distinct parts, namely Part A and Part B, which respectively depict dense and sparse crowd scenarios. Part A comprises a total of 482 images, with an average resolution of 589 × 868, featuring approximately 240,000 instances. These images were collected from various online sources. However, Part B contains 716 images, with a resolution of 768 × 1024, portraying a total of 88,488 individuals. The images in Part B were obtained from real-life shots of crowd scenes captured on the streets of Shanghai, China.

The *UCF-QNRF* [26] dataset consists of 1,535 images with an average resolution of 2013 × 2902. There are approximately 1.25 million instances, including some extremely crowded scenes.

The *UCF_CC_50* [25] dataset contains only 50 images with an average resolution of 2101 × 2888 but has about 64,000 instances, which is a highly crowded dataset.

Since the Swin transformer only supports fixed-size images, all datasets are cropped to a uniform size of 384 × 384. For datasets with varying original image sizes, we first resize them uniformly to 1152 × 768 and subsequently crop them to the desired dimensions. During training, we utilize horizontal flipping to augment the data and employ smooth L1 as the chosen loss function. The model is optimized using Adam [28], with a batch size of 32 and an initial learning rate of 1e-5, which decays to 1e-6 after 200 epochs. The experiments are conducted on an Ubuntu 20.04 platform equipped with an NVIDIA GeForce RTX 3090 (with ~24 GB of memory).

4.2 Evaluation Criteria

We evaluate the counting performance using **Mean Absolute Error (MAE)** and **Root Mean Squared Error (RMSE)**.

$$MAE = \frac{1}{m}\sum_{i=1}^{m} |y_i - \hat{y}_i|, \tag{7}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2}, \tag{8}$$

where m is the number of samples, y_i is the number of ground truth, and $\hat{y_i}$ is the predicted number for the i-th sample.

4.3 Performance Comparison and Analysis

We have conducted a comprehensive set of experiments on four publicly available datasets to demonstrate the effectiveness of our proposed approach. First, we compare the validation and test sets of JHU-CROWD++, which comprises the largest amount of data, as presented in Tables 1 and 2. Subsequently, we evaluate the performance of our method on the Shanghai Tech A/B, UCF-QNRF, and UCF_CC_50 datasets, as outlined in Table 3. It is pertinent to note that, to provide a more intuitive understanding of the performance of weakly supervised crowd counting methods, we have also compared our results with popular strongly supervised approaches, where the location information is specified in the label column of the table.

Our proposed HACC model demonstrates superior performance over current state-of-the-art methods on the JHU-CROWD++ dataset. In particular, both the val and test sets of JHU-CROWD++ are stratified into three distinct sub-datasets, each representing crowd scenes with varying densities. We evaluate the performance of our HACC model under different counting scenarios and report the results in Table 1. Additionally, we test our model on the entire dataset and also report the performance of the best model on the validation set in Table 2. Hence, Table 1 provides

Table 1. Quantitative Results on the JHU-CROWD++ (Val Set)

Method	Venue	Label		Low		Medium		High		Overall	
		L	N	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MCNN [81]	CVPR16	✓	✓	90.6	202.9	125.3	259.5	494.9	856.0	160.6	377.7
CMTL [58]	AVSS17	✓	✓	50.2	129.2	88.1	170.7	583.1	986.5	138.1	379.5
DSSI-Net [43]	ICCV19	✓	✓	50.3	85.9	82.4	164.5	436.6	814.0	116.6	317.4
CAN [45]	CVPR19	✓	✓	34.2	69.5	65.6	115.3	336.4	619.7	89.5	239.3
SANet [7]	ECCV18	✓	✓	13.6	26.8	50.4	78.0	397.8	749.2	82.1	272.6
CSRNet [36]	CVPR18	✓	✓	22.2	40.0	49.0	99.5	302.5	669.5	72.2	249.9
MBTTBF [60]	ICCV19	✓	✓	23.3	48.5	53.2	119.9	294.5	674.5	73.8	256.8
SFCN [71]	CVPR19	✓	✓	11.8	19.8	**39.3**	73.4	297.3	679.4	62.9	247.5
BL [52]	ICCV19	✓	✓	**6.9**	**10.3**	39.7	85.2	**279.8**	**620.4**	**59.3**	**229.2**
CG-DRCN [57]	TPAMI20	✓	✓	17.1	44.7	40.8	**71.2**	317.4	719.8	67.9	262.1
TC-Token [37]	SCIS22	✗	✓	7.1	10.7	33.3	54.6	302.5	557.4	58.4	201.1
TC-GAP [37]	SCIS22	✗	✓	6.7	9.5	34.5	55.8	285.9	532.8	56.8	193.6
HACC (ours)	–	✗	✓	**4.7**	**6.9**	**26.5**	**40.4**	**238.7**	**501.9**	**48.8**	**186.3**

The smaller the MAE and RMSE metrics, the better the performance. “L” and “N”indicate location and number, respectively. “Low,” “Medium,” and “High” respectively indicate three categories based on different ranges: [0,50], (50,500], and 500+.

Table 2. Quantitative Results on the JHU-CROWD++ (Test Set)

Method	Venue	Label		Low		Medium		High		Overall	
		L	N	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MCNN [81]	CVPR16	✓	✓	97.1	192.3	121.4	191.3	618.6	1,166.7	188.9	483.4
CMTL [58]	AVSS17	✓	✓	58.5	136.4	81.7	144.7	635.3	1,225.3	157.8	490.4
DSSI-Net [43]	ICCV19	✓	✓	53.6	112.8	70.3	108.6	525.5	1,047.4	133.5	416.5
CAN [45]	CVPR19	✓	✓	37.6	78.8	56.4	86.2	384.2	789.0	100.1	314.0
SANet [7]	ECCV18	✓	✓	17.3	37.9	46.8	69.1	397.9	817.7	91.1	320.4
CSRNet [36]	CVPR18	✓	✓	27.1	64.9	43.9	71.2	356.2	784.4	85.9	309.2
MBTTBF [60]	ICCV19	✓	✓	19.2	58.8	41.6	66.0	352.2	760.4	81.8	299.1
SFCN [71]	CVPR19	✓	✓	16.5	55.7	38.1	59.8	341.8	758.8	77.5	297.6
BL [52]	ICCV19	✓	✓	10.1	32.7	34.2	54.5	352.0	768.7	75.0	299.9
CG-DRCN [57]	TPAMI20	✓	✓	19.5	58.7	38.4	62.7	367.3	837.5	82.3	328.0
AutoScale [75]	IJCV21	✓	✓	13.2	30.2	32.3	52.8	425.6	916.5	85.6	356.1
D2CNet [12]	TIP21	✓	✓	12.6	38.5	36.5	56.3	330.3	748.6	73.7	292.5
GL [66]	CVPR21	✓	✓	–	–	–	–	–	–	59.9	259.5
TopoCount [1]	AAAI21	✓	✓	**8.2**	**20.5**	**28.9**	**50.0**	282.0	685.8	60.9	267.4
CLTR [38]	ECCV22	✓	✓	8.3	21.8	30.7	53.8	**265.2**	**614.0**	59.5	240.6
ChfL [56]	CVPR22	✓	✓	–	–	–	–	–	–	57.0	235.7
MAN [40]	CVPR22	✓	✓	–	–	–	–	–	–	**53.4**	**209.9**
TC-Token [37]	SCIS22	✗	✓	8.5	23.2	33.3	71.5	368.3	816.4	76.4	319.8
TC-GAP [37]	SCIS22	✗	✓	7.6	**16.7**	34.8	73.6	354.8	752.8	74.9	295.6
CrowdMLP [70]	Arxiv22	✗	✓	–	–	–	–	–	–	67.6	256.2
DMCNet [69]	WACV23	✗	✓	–	–	–	–	–	–	69.6	246.9
HACC (ours)	–	✗	✓	**6.8**	18.0	**28.9**	**46.0**	**299.2**	**632.8**	**63.2**	**246.7**

The smaller the MAE and RMSE metrics, the better the performance.

insights into the exceptional performance of our model under extreme density counting scenarios, whereas Table 2 presents the comprehensive performance of our model across all scenarios. Our HACC model’s performance on the UCF-QNRF, Shanghai Tech A/B, and UCF_CC_50 datasets is also presented in Table 3, which further corroborates the competitiveness of our approach.

Table 3. Performance of the HACC Model on the Three Datasets of Shanghai Tech A/B (SHTA and SHTB), UCF-QNRF, and UCF_CC_50

Method	Venue	Label		SHTA		SHTB		UCF-QNRF		UCF_CC_50	
		L	N	A	S	A	S	A	S	A	S
MCNN [81]	CVPR16	✓	✓	110.2	173.2	26.4	41.3	277.0	426.0	377.6	509.1
Switching [3]	CVPR17	✓	✓	90.4	135.4	21.6	33.4	–	–	318.1	439.2
CSRNet [36]	CVPR18	✓	✓	68.2	115.0	10.6	16.0	–	–	266.1	397.5
SFCN [71]	CVPR19	✓	✓	64.8	107.5	7.6	13.0	102.0	171.4	214.2	318.2
AEDN [80]	TOMM20	✓	✓	63.8	106.8	8.1	13.4	123.0	198.0	255.0	330.5
HyGNN [50]†	AAAI20	✓	✓	60.2	94.5	7.5	12.7	100.8	185.3	184.4	270.1
TopoCount [1]	AAAI21	✓	✓	56.9	95.2	6.5	10.6	87.3	142.4	–	–
P2PNet [82]	ICCV21	✓	✓	52.7	85.1	6.3	**9.9**	85.3	154.5	172.7	256.2
D2CNet [12]	TIP21	✓	✓	57.2	93.0	6.3	10.7	81.7	137.5	182.1	254.9
SASNet [62]	AAAI21	✓	✓	53.6	88.4	6.4	**9.9**	85.2	147.3	**161.4**	**234.5**
CCTrans [48]	Arxiv21	✓	✓	**52.3**	**84.9**	**6.2**	**9.9**	82.8	142.3	168.7	**234.5**
FIDTM [39]	TMM22	✓	✓	57.0	103.4	6.9	11.8	89.0	153.5	–	–
CLTR [38]	ECCV22	✓	✓	61.2	104.6	7.8	13.7	89.0	159.0	184.1	258.3
MAN [40]	CVPR22	✓	✓	56.8	90.3	–	–	**77.3**	**131.5**	–	–
ChfL [56]	CVPR22	✓	✓	57.5	94.3	6.9	11.0	80.3	137.6	–	–
Yang et al. [77]	ECCV20	✗	✓	–	–	12.3	21.2	104.6	**145.2**	–	–
MATT [31]	PR21	✗	✓	80.1	129.4	11.7	17.5	–	–	355.0	550.2
CCTrans [65]	Arxiv21	✗	✓	64.4	95.4	**7.0**	**11.5**	92.1	158.9	245.0	343.0
TC-GAP [37]	SCIS22	✗	✓	66.1	105.1	9.3	16.1	97.2	168.5	–	–
CrowdFormer [55]	arxiv22	✗	✓	62.1	94.8	8.5	13.6	93.3	160.9	229.6	360.3
JCTNet [68]	Arxiv22	✗	✓	62.8	95.6	7.2	**11.5**	**90.0**	161.0	222.9	306.5
DMCNet [69]	WACV23	✗	✓	58.5	**84.6**	8.6	13.7	96.5	164.0	–	–
HACC (ours)	–	✗	✓	**58.3**	**84.6**	7.5	11.8	92.9	168.7	**220.4**	**338.2**

The smaller the MAE and RMSE metrics, the better the performance. To be more intuitive, we **bold** and underline the best and second best performing models, respectively. The method based on the GNN is marked with a dagger (†). "A" and "S" denote MAE and RMSE, respectively.

To analyze the performance of the HACC model, we distinguish between dense and sparse counting scenarios. The HACC model shows outstanding performance on crowded datasets, such as Shanghai Tech A, JHU-CROWD++ (high sub-dataset), UCF-QNRF, and UCF_CC_50. This remarkable performance is mainly attributed to the MDP module and the HA module. The MDP module uses multiple branch networks with different dilated rates, allowing the model to effectively capture small-scale counted objects in dense scenes. Additionally, the output of the branch with a dilated rate of 1 is superimposed onto the other three branches to further strengthen the underlying feature information. Furthermore, the HA module enables the model to perceive densely crowded areas in the counting scene, thereby improving the accuracy of counting in crowded areas, which is particularly evident in dense datasets. To intuitively understand the contribution of the HA module, we visualize the attention heatmap of the network in Section 5. On datasets with lower density, such as JHU-CROWD++ (low and medium subsets) and Shanghai Tech B, our model does not exhibit significant advantages in performance. We infer that the distance among individuals in low-density datasets is relatively large, making it challenging for the HA module to establish relatively long-distance associations.

4.4 Ablation Experiments

To assess the individual contribution of each module and its key components to the HACC model, we conducted ablation experiments on Shanghai Tech A/B. First, we evaluated the effectiveness

Table 4. Ablation Experiments of the HACC Model on Shanghai Tech A/B (SHTA and SHTB) Mainly Include the MDP Module and Its BS Operation and Hypergraph (Association) Module

Bone	MDP	BS	HGNN	HA	SHTA		SHTB	
					MAE	RMSE	MAE	RMSE
✓					64.3	99.5	9.4	15.4
✓	✓				63.2	99.9	8.3	13.9
✓	✓	✓			62.8	98.8	8.2	13.5
✓	✓	✓	✓		61.7	96.6	8.0	12.7
✓	✓	✓		✓	**58.3**	**84.6**	**7.5**	**11.8**

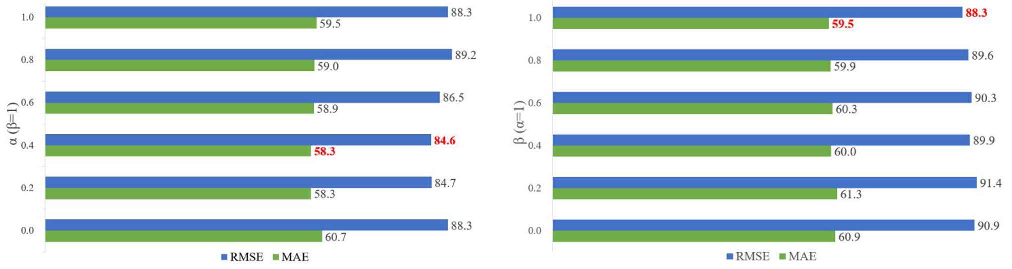
of the MDP module and the **Branch Stacking (BS)** with a dilated rate of 1 onto the other three branches. Subsequently, we examined the contribution of the HA module. To distinguish the performance between the HA module and the traditional HGNN, we conducted experiments by replacing the HA module with HGNN while keeping other parameters constant during training.

After conducting ablation experiments on the MDP module and the HA module separately at Shanghai Tech A/B, we obtained the results shown in Table 4. The results indicate that the MDP module and its BS operations can improve the performance of the HACC model. Moreover, we observe that introducing the traditional HGNN does improve the counting performance, albeit to a limited extent. However, the performance improvement of the HA module is significant on Shanghai Tech A, but only slight on Shanghai Tech B. This result aligns with our theory that dense individuals are more likely to establish associations among features.

To further investigate the effectiveness of the HA module, we conducted ablation experiments on some hyperparameters in the hypergraph construction process, including α and β in Equation (4). The hyperparameters α and β determine the relative weights of the distance matrix M_{dis} and the learnable similarity correlation matrix M_{sim}, which greatly influence the basis for hypergraph construction. To fully explore the effect of these parameters, we varied one parameter from 0 to 1 while keeping the other at 1. The experimental results are shown in Figure 6. Our observations are as follows: (1) the performance in Figure 6(a) is generally better than that in Figure 6(b), indicating that the larger the weight of M_{sim}, the better the performance; (2) when either α or β is zero, the performance decreases significantly, indicating that the fusion of M_{dis} and M_{sim} outperforms using a single matrix; and (3) the best counting performance is achieved when $\alpha = 0.4$ and $\beta = 1$. Furthermore, we conducted ablation experiments on the number of neighboring nodes in the hyperedge construction process, and the results on the Shanghai Tech A dataset are shown in Table 5. As the number of nodes increases, a single hyperedge can better capture local features, thereby increasing its characterization ability and leading to progressively stronger counting performance of the model.

4.5 Computational Cost

The computational cost of a model is a crucial factor that affects its practical application. Typically, this factor is evaluated by considering the number of GFLOPs and model parameters. In this study, we have selected recent methods for comparison and present the results in Table 6. As can be observed, the overall computational cost of our HACC model is relatively small, making it suitable for further applications. First, the computational cost of CCST and DCST is comparable to our HACC model. This similarity is because we have all adopted the expensive Swin transformer as the backbone of our models. Second, due to the multi-branch design, our MDP module is also slightly expensive in computation. Finally, the computational cost of the HA module mainly arises from



(a) (b)

Fig. 6. Ablation experiments of hyperparameters affecting the relative weights of the distance matrix M_{dis} and the learnable similarity correlation matrix M_{sim}. (a) Variation in counting performance as α changes from 0 to 1 when $\beta = 1$. (b) Variation in counting performance as β changes from 0 to 1 when $\alpha = 1$.

Table 5. Ablation Experiments on the Number of Neighboring Nodes in the Hyperedge Construction Process

	3	6	9	12	15
MAE	58.9	58.6	58.3	**58.2**	58.3
RMSE	85.9	85.2	**84.6**	85.6	**84.6**

Table 6. Comparison of the Computational Cost of Different Models

	CCST	DCST	SASNet	P2PNet	GAP	**HACC**
GFLOPs	323.6	154.8	130.9	58.8	49.3	**103.9**
Params	294.7	252.3	38.9	19.2	89.1	**199.0**

We measure GFLOPs with 384 × 384 resolution images as input. Works compared to HACC (we have bolded it) include CCST [33], DCST [19], SASNet [62], P2PNet [82], and TransCrowd-GAP (GAP) [37].

the calculation of the nearest neighbors. However, the overall cost of the HA module is relatively small, making it ideal for future research applications.

5 DISCUSSION ON THE HA MODULE

To gain a better understanding of the HA module, we conducted further analysis. We visualized the attention heatmap to demonstrate the effect of the HA module on scene understanding, as presented in Figure 7. The visualization reveals that our HA module helps the model perceive the distribution of crowds in the scene, which is advantageous for scene understanding. Furthermore, as the preceding scene understanding effect is similar to that of the attention mechanism, we compared the effect of the HA module with that of the attention mechanism. The visualization of the heatmap is illustrated in Figure 8, and the model performance is shown in Table 6.

5.1 Attention Heatmap

To gain an intuitive understanding of the effect of the HA module, we visualized the network attention heatmaps before and after the module. Additionally, we trained a counting model based on the traditional HGNN while keeping other conditions constant to verify the improvement of our HA module relative to the HGNN. The results are presented in Figure 7. The first row of the three rows shows the heatmap before the hypergraph module, the middle row depicts the heatmap of HGNN, and the last row displays the heatmap of our proposed HA. Two conclusions can be drawn

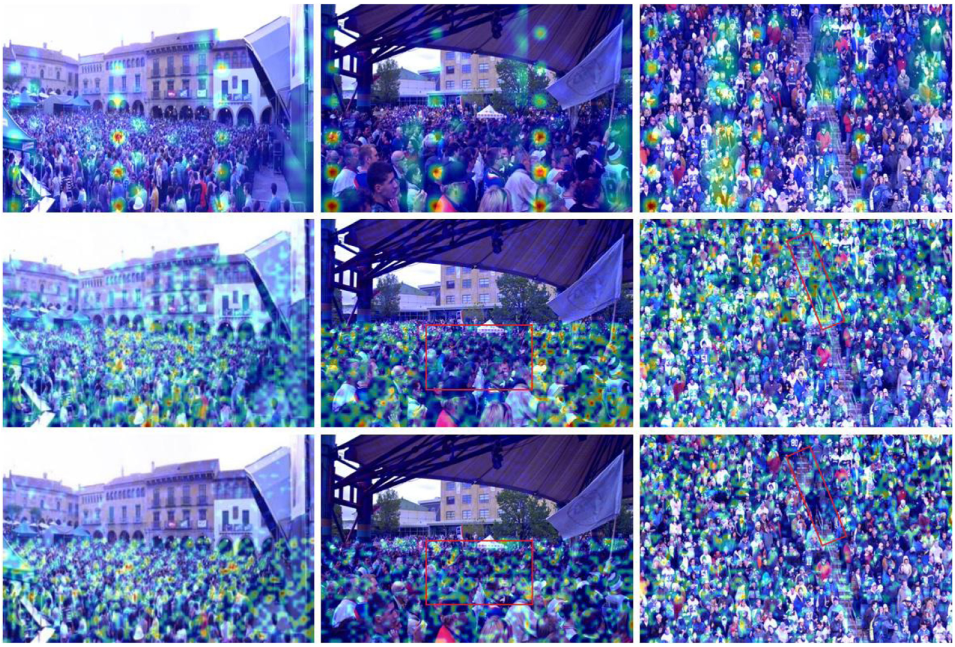


Fig. 7. The visualization results of the attention heatmap before and after the hypergraph (association) module. The first row is the heatmap before the hypergraph module, the middle is the heatmap of the HGNN, the last is the heatmap of HA. Best viewed in red boxes.

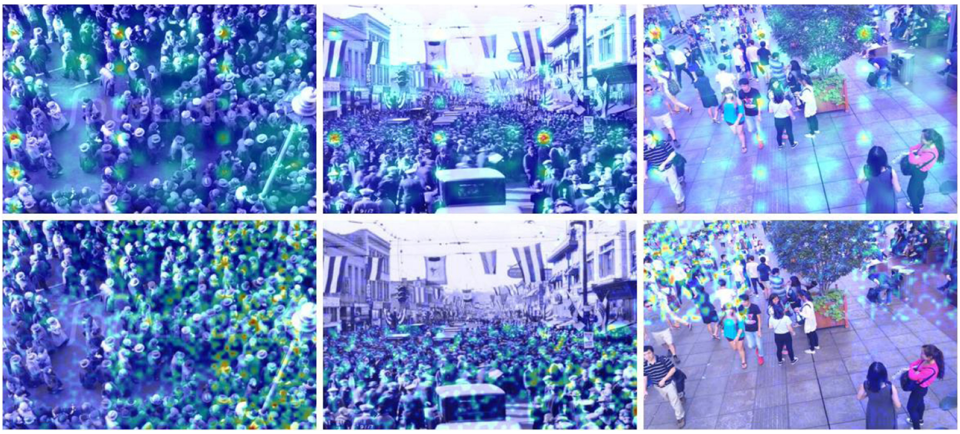


Fig. 8. Comparison of heatmap visualization results generated by the attention module CBAM (the top row) and the HA module (the following row).

from the observations. First, HGNN enables the model to perceive the approximate distribution of crowds in the counting scene. Second, compared to HGNN, our proposed HA module significantly improves the model's ability to understand the scene, leading to better counting performance.

The effect of the HA module is strikingly similar to the way humans perceive an image. When we observe an image for crowd counting, our attention is primarily drawn to the part of the crowd in the image, enabling us to see more clearly.

Table 7. Comparison of HA Module and Attention Mechanisms on the Shanghai Tech A/B (SHTA and SHTB) Datasets

Module	Label		SHTA		SHTB	
	Location	Number	MAE	RMSE	MAE	RMSE
Base	✗	✓	62.8	98.8	8.2	13.5
Base+SE	✗	✓	63.6	99.5	9.2	14.0
Base+GAM	✗	✓	62.2	96.8	8.1	12.5
Base+CBAM	✗	✓	61.7	97.1	7.8	12.0
Base+HA	✗	✓	**58.3**	**84.6**	**7.5**	**11.8**

Base refers to the combination of the backbone network and MDP module.

5.2 Compared with the Attention Mechanism

Given that the effect of the HA module is similar to the attention mechanism, we compared these two modules. Attention mechanisms have achieved significant success in various vision tasks, such as object detection [11], semantic segmentation [16], and 3D vision [49]. However, for the weakly supervised crowd counting task, where the supervision information is only a number, can the attention mechanism effectively perceive the crowded areas in the counting scene?

To answer this question, we replaced the HA module with three popular attention mechanisms, namely SE [24], CBAM [73], and GAM [46]. The SE module adaptively recalibrates channel feature responses by explicitly modeling interdependencies among channels. CBAM uses channel attention and spatial attention in turn, and the obtained weights are multiplied with the input feature maps for adaptive feature refinement. GAM is further designed on the basis of CBAM, dedicated to improving deep neural network performance by reducing information dispersion and amplifying global interactive representations. We kept other parameters unchanged, recorded the model's performance, and visualized the attention heatmap. The results are presented in Table 7 and Figure 8, respectively.

Our experiments show that compared with the three attention modules mentioned previously, our HA module achieves better performance on the crowded Shanghai Tech A/B datasets. Further, the attention heatmaps in Figure 8 demonstrate that the attention module does not effectively perceive the distribution information of the crowd in the counting scene. However, the HA module can effectively model the crowd distribution and achieve scene understanding.

6 CONCLUSION AND OUTLOOK

This article presented the HACC model, which is based on the HGNN and is designed for the weakly supervised crowd counting task. The model incorporates a Swin transformer based backbone network to capture both global contextual and local fine-grained information. To enhance the multi-scale representation of the scene, we introduced a new MDP module. Additionally, we proposed a HA module to leverage the higher-order associations among features and realize scene understanding. Our experimental results showed that the proposed HA module is able to effectively extract the distribution of crowd density using count-level supervision information alone. This highlights the potential of hypergraph-based approaches as a new direction for addressing the uneven distribution of crowd density.

The article also proposed the HA module, demonstrating its enormous potential in weakly supervised counting tasks. However, the current HA module still faces some difficulty in modeling complex crowd distributions, and the HACC model we introduced in this article is relatively cumbersome. In the future, we will strive to propose more lightweight and efficient methods to enable more widespread application in practical scenarios. Additionally, based on the characteristic of node aggregation in the hypergraph module, we expect it to perform well in point-supervised tasks, such as crowd localization, which will also be a focus of our future work.

REFERENCES

[1] Shahira Abousamra, Minh Hoai, Dimitris Samaras, and Chao Chen. 2021. Localization in the crowd with topological constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 872–881.

[2] James Atwood and Don Towsley. 2016. Diffusion-convolutional neural networks. In *Proceedings of Advances in Neural Information Processing Systems*, Vol. 29.

[3] Deepak Babu Sam, Shiv Surya, and R. Venkatesh Babu. 2017. Switching convolutional neural network for crowd counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.* 5744–5752.

[4] Walid Balid, Hasan Tafish, and Hazem H. Refai. 2017. Intelligent vehicle counting and classification sensor for real-time traffic surveillance. *IEEE Transactions on Intelligent Transportation Systems* 19, 6 (2017), 1784–1794.

[5] Lokesh Boominathan, Srinivas S. S. Kruthiventi, and R. Venkatesh Babu. 2016. CrowdNet: A deep convolutional network for dense crowd counting. In *Proceedings of the ACM International Conference on Multimedia.* 640–644.

[6] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. 2018. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision.* 734–750.

[7] Xinkun Cao, Zhipeng Wang, Yanyun Zhao, and Fei Su. 2018. Scale aggregation network for accurate and efficient crowd counting. In *Proceedings of the European Conference on Computer Vision.* 734–750.

[8] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. 2017. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 4 (2017), 834–848.

[9] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

[10] Xinya Chen, Yanrui Bin, Changxin Gao, Nong Sang, and Hao Tang. 2020. Relevant region prediction for crowd counting. *Neurocomputing* 407 (2020), 399–408.

[11] Xiaoshuang Chen and Hongtao Lu. 2022. Reinforcing local feature representation for weakly-supervised dense crowd counting. *arXiv preprint arXiv:2202.10681* (2022).

[12] Jian Cheng, Haipeng Xiong, Zhiguo Cao, and Hao Lu. 2021. Decoupled two-stage crowd counting and beyond. *IEEE Transactions on Image Processing* 30 (2021), 2862–2875.

[13] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Proceedings of Advances in Neural Information Processing Systems*, Vol. 29.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[15] Thorsten Falk, Dominic Mai, Robert Bensch, Özgün Çiçek, Ahmed Abdulkadir, Yassine Marrakchi, Anton Böhm, et al. 2019. U-Net: Deep learning for cell counting, detection, and morphometry. *Nature Methods* 16, 1 (2019), 67–70.

[16] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. 2020. Few-shot object detection with attention-RPN and multi-relation detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 4013–4022.

[17] Yuchun Fang, Yifan Li, Xiaokang Tu, Taifeng Tan, and Xin Wang. 2020. Face completion with hybrid dilated convolution. *Signal Processing: Image Communication* 80 (2020), 115664.

[18] Yifan Feng, Haoxuan You, Zizhao Zhang, Rongrong Ji, and Yue Gao. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence.* 3558–3565.

[19] Junyu Gao, Maoguo Gong, and Xuelong Li. 2021. Congested crowd instance localization with dilated convolutional Swin transformer. *arXiv preprint arXiv:2108.00584* (2021).

[20] Junyu Gao, Qi Wang, and Yuan Yuan. 2019. SCAR: Spatial-/channel-wise attention regression networks for crowd counting. *Neurocomputing* 363 (2019), 1–8.

[21] Jason M. Grant and Patrick J. Flynn. 2017. Crowd scene understanding from video: A survey. *ACM Transactions on Multimedia Computing, Communications, and Applications* 13, 2 (2017), 1–23.

[22] Dongyan Guo, Yanyan Shao, Ying Cui, Zhenhua Wang, Liyan Zhang, and Chunhua Shen. 2021. Graph attention tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.* 9543–9552.

[23] Shenghua He, Kyaw Thu Minn, Lilianna Solnica-Krezel, Mark A. Anastasio, and Hua Li. 2021. Deeply-supervised density regression for automatic cell counting in microscopy images. *Medical Image Analysis* 68 (2021), 101892.

[24] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops.* 7132–7141.

[25] Haroon Idrees, Imran Saleemi, Cody Seibert, and Mubarak Shah. 2013. Multi-source multi-scale counting in extremely dense crowd images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2547–2554.

[26] Haroon Idrees, Muhmmad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Maadeed, Nasir Rajpoot, and Mubarak Shah. 2018. Composition loss for counting, density map estimation and localization in dense crowds. In *Proceedings of the European Conference on Computer Vision*. 532–546.

[27] Jianwen Jiang, Yuxuan Wei, Yifan Feng, Jingxuan Cao, and Yue Gao. 2019. Dynamic hypergraph neural networks. In *Proceedings of the International Joint Conference on Artificial Intelligence*. 2635–2641.

[28] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[29] Thomas N. Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[30] Issam H. Laradji, Negar Rostamzadeh, Pedro O. Pinheiro, David Vazquez, and Mark Schmidt. 2018. Where are the blobs: Counting by localization with point supervision. In *Proceedings of the European Conference on Computer Vision*. 547–562.

[31] Yinjie Lei, Yan Liu, Pingping Zhang, and Lingqiao Liu. 2021. Towards using count-level weak supervision for crowd counting. *Pattern Recognition* 109 (2021), 107616.

[32] Victor Lempitsky and Andrew Zisserman. 2010. Learning to count objects in images. In *Proceedings of Advances in Neural Information Processing Systems*, Vol. 23.

[33] Bo Li, Yong Zhang, Haihui Xu, and Baocai Yin. 2022. CCST: Crowd counting with swin transformer. *Visual Computer* 2022 (2022), 1–12.

[34] Wang Li, Huailin Zhao, Zhen Nie, and Yaoyao Li. 2020. Graph-based global reasoning network for crowd counting. In *Proceedings of the International Conference on Artificial Life and Robotics*.

[35] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. 2019. Scale-aware trident networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6054–6063.

[36] Yuhong Li, Xiaofan Zhang, and Deming Chen. 2018. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 1091–1100.

[37] Dingkang Liang, Xiwu Chen, Wei Xu, Yu Zhou, and Xiang Bai. 2022. TransCrowd: Weakly-supervised crowd counting with transformers. *Science China Information Sciences* 65, 6 (2022), 1–14.

[38] Dingkang Liang, Wei Xu, and Xiang Bai. 2022. An end-to-end transformer model for crowd localization. In *Proceedings of the European Conference on Computer Vision*. 38–54.

[39] Dingkang Liang, Wei Xu, Yingying Zhu, and Yu Zhou. 2022. Focal inverse distance transform maps for crowd localization. *IEEE Transactions on Multimedia*. Early access, September 2, 2022.

[40] Hui Lin, Zhiheng Ma, Rongrong Ji, Yaowei Wang, and Xiaopeng Hong. 2022. Boosting crowd counting via multifaceted attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19628–19637.

[41] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2117–2125.

[42] Liang Liu, Hao Lu, Haipeng Xiong, Ke Xian, Zhiguo Cao, and Chunhua Shen. 2019. Counting objects by blockwise classification. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 10 (2019), 3513–3527.

[43] Lingbo Liu, Zhilin Qiu, Guanbin Li, Shufan Liu, Wanli Ouyang, and Liang Lin. 2019. Crowd counting with deep structured scale integration network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1774–1783.

[44] Songtao Liu and Di Huang. 2018. Receptive field block net for accurate and fast object detection. In *Proceedings of the European Conference on Computer Vision*. 385–400.

[45] Weizhe Liu, Mathieu Salzmann, and Pascal Fua. 2019. Context-aware crowd counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 5099–5108.

[46] Yichao Liu, Zongru Shao, and Nico Hoffmann. 2021. Global attention mechanism: Retain information to enhance channel-spatial interactions. *arXiv preprint arXiv:2112.05561* (2021).

[47] Yuting Liu, Miaojing Shi, Qijun Zhao, and Xiaofang Wang. 2019. Point in, box out: Beyond counting persons in crowds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6469–6478.

[48] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 10012–10022.

[49] Zhe Liu, Xin Zhao, Tengteng Huang, Ruolan Hu, Yu Zhou, and Xiang Bai. 2020. TANet: Robust 3D object detection from point clouds with triple attention. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 11677–11684.

[50] Ao Luo, Fan Yang, Xin Li, Dong Nie, Zhicheng Jiao, Shangchen Zhou, and Hong Cheng. 2020. Hybrid graph neural networks for crowd counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 11693–11700.

[51] Zhongtian Ma, Zhiguo Jiang, and Haopeng Zhang. 2021. Hyperspectral image classification using feature fusion hypergraph convolution neural network. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2021), 1–14.

[52] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. 2019. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6142–6151.

[53] Daniel Onoro-Rubio and Roberto J. López-Sastre. 2016. Towards perspective-free object counting with deep learning. In *Proceedings of the European Conference on Computer Vision*. 615–629.

[54] Deepak Babu Sam, Neeraj N. Sajjan, R. Venkatesh Babu, and Mukundhan Srinivasan. 2018. Divide and grow: Capturing huge diversity in crowd images with incrementally growing CNN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 3618–3626.

[55] Siddharth Singh Savner and Vivek Kanhangad. 2022. CrowdFormer: Weakly-supervised crowd counting with improved generalizability. *arXiv preprint arXiv:2203.03768* (2022).

[56] Weibo Shu, Jia Wan, Kay Chen Tan, Sam Kwong, and Antoni B. Chan. 2022. Crowd counting in the frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19618–19627.

[57] Vishwanath Sindagi, Rajeev Yasarla, and Vishal M. M. Patel. 2022. JHU-CROWD++: Large-scale crowd counting dataset and a benchmark method. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 6 (2022), 2594–2609.

[58] Vishwanath A. Sindagi and Vishal M. Patel. 2017. CNN-based cascaded multi-task learning of high-level prior and density estimation for crowd counting. In *Proceedings of the IEEE International Conference on Advanced Video and Signal Based Surveillance*. 1–6.

[59] Vishwanath A. Sindagi and Vishal M. Patel. 2017. Generating high-quality crowd density maps using contextual pyramid CNNs. In *Proceedings of the IEEE International Conference on Computer Vision*. 1861–1870.

[60] Vishwanath A. Sindagi and Vishal M. Patel. 2019. Multi-level bottom-top and top-bottom feature fusion for crowd counting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1002–1012.

[61] Vishwanath A. Sindagi, Rajeev Yasarla, and Vishal M. Patel. 2019. Pushing the frontiers of unconstrained crowd counting: New dataset and benchmark method. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 1221–1231.

[62] Qingyu Song, Changan Wang, Yabiao Wang, Ying Tai, Chengjie Wang, Jilin Li, Jian Wu, and Jiayi Ma. 2021. To choose or to fuse? Scale selection for crowd counting. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 2576–2583.

[63] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2818–2826.

[64] Mengxiao Tian, Hao Guo, and Chengjiang Long. 2021. Multi-level attentive convolutional neural network for crowd counting. *arXiv preprint arXiv:2105.11422* (2021).

[65] Ye Tian, Xiangxiang Chu, and Hongpeng Wang. 2021. CCTrans: Simplifying and improving crowd counting with transformer. *arXiv preprint arXiv:2109.14483* (2021).

[66] Jia Wan, Ziquan Liu, and Antoni B. Chan. 2021. A generalized loss function for crowd counting and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 1974–1983.

[67] Chuan Wang, Hua Zhang, Liang Yang, Si Liu, and Xiaochun Cao. 2015. Deep people counting in extremely dense crowds. In *Proceedings of the ACM International Conference on Multimedia*. 1299–1302.

[68] Fusen Wang, Kai Liu, Fei Long, Nong Sang, Xiaofeng Xia, and Jun Sang. 2022. Joint CNN and transformer network via weakly supervised learning for efficient crowd counting. *arXiv preprint arXiv:2203.06388* (2022).

[69] Mingjie Wang, Hao Cai, Yong Dai, and Minglun Gong. 2023. Dynamic mixture of counter network for location-agnostic crowd counting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 167–177.

[70] Mingjie Wang, Jun Zhou, Hao Cai, and Minglun Gong. 2022. CrowdMLP: Weakly-supervised crowd counting via multi-granularity MLP. *arXiv preprint arXiv:2203.08219* (2022).

[71] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. 2019. Learning from synthetic data for crowd counting in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 8198–8207.

[72] Xiaolong Wang and Abhinav Gupta. 2018. Videos as space-time region graphs. In *Proceedings of the European Conference on Computer Vision*. 399–417.

[73] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. CBAM: Convolutional block attention module. In *Proceedings of the European Conference on Computer Vision*. 3–19.

[74] Peng Wu, Jing Liu, Yujia Shi, Yujia Sun, Fangtao Shao, Zhaoyang Wu, and Zhiwei Yang. 2020. Not only look, but also listen: Learning multimodal violence detection under weak supervision. In *Proceedings of the European Conference on Computer Vision*. 322–339.

[75] Chenfeng Xu, Dingkang Liang, Yongchao Xu, Song Bai, Wei Zhan, Xiang Bai, and Masayoshi Tomizuka. 2022. Autoscale: Learning to scale for crowd counting. *International Journal of Computer Vision* 130, 2 (2022), 405–434.

[76] Jianxing Yang, Yuan Zhou, and Sun-Yuan Kung. 2018. Multi-scale generative adversarial networks for crowd counting. In *Proceedings of the International Conference on Pattern Recognition*. 3244–3249.

[77] Yifan Yang, Guorong Li, Zhe Wu, Li Su, Qingming Huang, and Nicu Sebe. 2020. Weakly-supervised crowd counting learns from sorting rather than locations. In *Proceedings of the European Conference on Computer Vision*. 1–17.

[78] Lingke Zeng, Xiangmin Xu, Bolun Cai, Suo Qiu, and Tong Zhang. 2017. Multi-scale convolutional neural networks for crowd counting. In *Proceedings of the IEEE International Conference on Image Processing*. 465–469.

[79] Qiang Zhai, Fan Yang, Xin Li, Guo-Sen Xie, Hong Cheng, and Zicheng Liu. 2022. Co-communication graph convolutional network for multi-view crowd counting. *IEEE Transactions on Multimedia*. Early access, August 17, 2022.

[80] Anran Zhang, Xiaolong Jiang, Baochang Zhang, and Xianbin Cao. 2020. Multi-scale supervised attentive encoder-decoder network for crowd counting. *ACM Transactions on Multimedia Computing, Communications, and Applications* 16, 1s (2020), 1–20.

[81] Yingying Zhang, Desen Zhou, Siqin Chen, Shenghua Gao, and Yi Ma. 2016. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 589–597.

[82] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 6881–6890.

[83] Liang Zhu, Zhijian Zhao, Chao Lu, Yining Lin, Yao Peng, and Tangren Yao. 2019. Dual path multi-scale fusion networks with attention for crowd counting. *arXiv preprint arXiv:1902.01115* (2019).

[84] Zhikang Zou, Yu Cheng, Xiaoye Qu, Shouling Ji, Xiaoxiao Guo, and Pan Zhou. 2019. Attend to count: Crowd counting with adaptive capacity multi-scale CNNs. *Neurocomputing* 367 (2019), 75–83.

Received 12 December 2022; revised 18 March 2023; accepted 20 April 2023