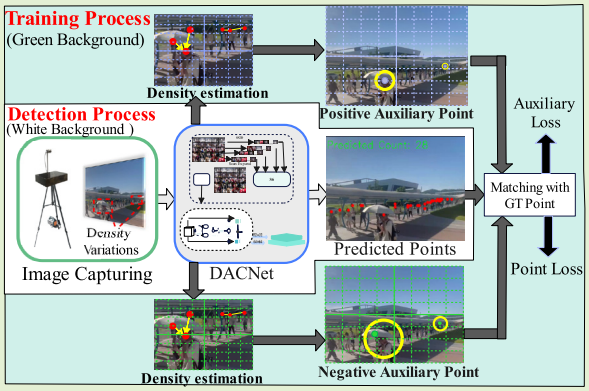
DACNet: A Density-Adaptive Counting Network for Real-World Crowd Analysis Without Overhead

Jianping Yue, Bohuan Xue, *Member, IEEE*, Wenli Wu, Rui Fan, *Senior Member, IEEE*, and Xiaoyu Tang, *Member, IEEE*

***Abstract*—In practical applications of crowd counting, the density and scale of human heads often vary significantly due to the influence of the camera's perspective effect. Point-based methods fail to consider crowd density variations and face issues with inaccurate matching. Inspired by the spatial perception function of the posterior parietal cortex in the human brain, this article proposes a density-adaptive counting network (DACNet), which assists object counting through auxiliary points. First, we propose a lightweight detail enhancement Mamba block (DEmamba Block), which combines convolution and state space models (SSMs) to enhance blurred details in densely crowded regions. Second, we propose a plug-and-play adaptive channel focus module (ACFM). ACFM introduces a channel weight selection algorithm, leveraging the advantages of multiple weights. Finally, we propose a density-adaptive auxiliary point guidance (DA-APG) strategy in the detection head. DA-APG generates positive and negative auxiliary points at varying distances around the ground truth points based on crowd density as additional supervisory signals, addressing the issue of crowd density variation. Moreover, this DA-APG strategy is only applied during training, and does not incur additional computational cost. To facilitate research on crowd density variations in real-world scenarios, we introduce a specialized dataset named VariDensity-CC. Experiments on nine datasets show that DACNet achieves the best overall balance between accuracy and speed. Furthermore, DACNet has been deployed on edge computing devices for real-world testing and demonstrates real-time performance. The code and dataset are available at: https://github.com/SCNU-RISLAB/DACNet**



***Index Terms*— Crowd counting, density-adaptive, edge computing, object counting, sensor applications.**

I. Introduction

OBJECT counting, especially crowd counting, has numerous real-world applications [1], [2], [3]. The primary objective of crowd counting is to estimate the number of people in a given image [4], [5], [6], while a supplementary task, known as crowd localization, facilitates subsequent crowd behavior analysis in various scenarios. Currently, there are three basic methods for crowd counting: detection-based methods [7], map-based methods [8], and point-based methods [7]. Among them, the point-based method directly predicts the location and number of points. With its elegant end-to-end process and excellent localization ability, it is well-suited for real-world scenarios. Therefore, this article selects the point-based model P2PNet [9] as the baseline model for real-world application scenarios.

However, in real-world applications, variations in crowd density and scale are almost inevitable due to factors such as camera height and shooting angles. We observed that the density variations in crowd images often lead to mismatches between point proposals and ground truth points, limiting the application of point-based methods in real-world scenarios. Especially in high-density scenarios, due to the extreme proximity between points, point proposals are highly likely to match with incorrect ground truth points. As shown in

Received 22 September 2025; accepted 30 September 2025. Date of publication 5 November 2025; date of current version 30 December 2025. The associate editor coordinating the review of this article and approving it for publication was Prof. Chinmay Chakraborty. *(Corresponding author: Bohuan Xue.)*

Jianping Yue, Bohuan Xue, and Wenli Wu are with the School of Data Science and Engineering, Xingzhi College, South China Normal University, Shanwei 516600, China (e-mail: 20228132026@m.scnu.edu.cn; bxue@ieee.org; 20238132076@m.scnu.edu.cn).

Rui Fan is with the College of Electronic and Information Engineering, Shanghai Institute of Intelligent Science and Technology, Shanghai Research Institute for Intelligent Autonomous Systems, Shanghai Key Laboratory of Intelligent Autonomous Systems, State Key Laboratory of Autonomous Intelligent Unmanned Systems, and the Frontiers Science Center for Intelligent Autonomous Systems (Ministry of Education), Tongji University, Shanghai 201804, China (e-mail: rui.fan@ieee.org).

Xiaoyu Tang is with the School of Electronic and Information Engineering and Xingzhi College, South China Normal University, Shanwei 516600, China (e-mail: tangxy@scnu.edu.cn).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSEN.2025.3625467, provided by the authors.

Digital Object Identifier 10.1109/JSEN.2025.3625467

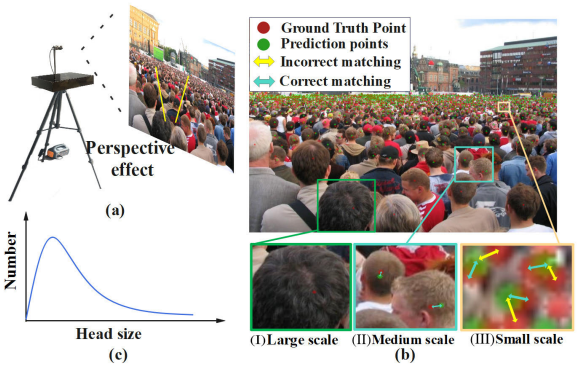


Fig. 1. Challenges in real-world crowd counting applications. (a) Perspective distortion in real scenes causes pronounced variations in head scale across the image. (b) Large foreground heads can be difficult to recognize, while densely packed background heads often suffer from mismatches and blurred boundaries. (c) Numerous small-scale heads are heavily occluded and low-resolution, making them difficult to localize accurately.

Fig. 1(b), in small-scale regions, a single predicted point may correspond to multiple ground truth points, further exacerbating the matching errors. In addition, point-based methods often suffer from limited feature representations and complex background interference, which further constrain counting accuracy. This is especially evident in densely crowded regions, where individual heads offer minimal distinguishable features, as shown in Fig. 1(b). The majority of heads are concentrated at small scales, as shown in Fig. 1(c). Therefore, effectively handling small-scale heads is key to improving the model's accuracy. Despite these challenges, there has been little systematic research addressing the limitations caused by density and scale variations in point-based crowd counting.

Variations in crowd density and scale are almost inevitable in real-world scenarios [10], which limits the practical application of point-based methods. Existing solutions include introducing additional annotated points [11], bounding boxes [11], or generating candidate regions based on scale variations [12]. While these methods have achieved some success, they often incur additional computational overhead and are not always compatible with point-based approaches [13]. Under such conditions, leveraging auxiliary information for object counting without increasing computational complexity becomes a promising alternative.

In the human brain, object recognition relies not only on prior representations but also on spatial relationships with surrounding objects, which are processed by the posterior parietal cortex. Inspired by this cognitive mechanism, we propose density-adaptive counting network (DACNet) that addresses the challenges posed by density variation by generating auxiliary points to enhance point-based crowd counting performance. First, we propose a lightweight detail enhancement Mamba block (DEmamba Block), which combines convolution and state space models (SSMs) to enhance blurred details in densely crowded regions. Second, we propose a plug-and-play adaptive channel focus module (ACFM) to tackle crowd density variations and background noise. Finally, to address the issues of crowd density variation in real-world scenarios, we propose a density-adaptive auxiliary point guidance (DA-APG) strategy. The main contributions of this study can be summarized as follows.

1) We propose an end-to-end real-time crowd counting network along with a dataset, VariDensity-CC, specifically designed for crowd density variations. Extensive experiments are conducted on nine datasets. Notably, we deployed the model on an edge computing device and monitored campus pedestrian flow, which can be applied in scenarios such as campus surveillance and accident prevention.
2) We propose a lightweight DEmamba Block that combines convolutional operations with an SSM to enhance blurred details in densely crowded regions. By integrating a small-kernel design with an enhanced Mamba architecture, DEmamba enables efficient extraction of fine-grained details in dense regions while maintaining low computational complexity.
3) We propose a plug-and-play ACFM. The channel selection algorithm in ACFM can leverage the advantages of different weights, filling the gap in integrating channel weights with varying characteristics.
4) A DA-APG strategy is introduced in the detection head. By generating positive and negative auxiliary points at varying distances around the true target points as additional supervisory signals, DA-APG solves the long-standing problem of crowd density variation in real-world application scenarios.

II. Related Work

A. Crowd Counting Methodologies

In recent years, object counting methods have shifted from traditional sensor-based approaches [14], [15] to deep learning-based methods [16], [17], [18]. Traditional methods using Wi-Fi signals, radar [2], [14], [19], [20], [21], [22], or visual image processing struggle in dense scenes. As a result, crowd counting has increasingly turned to computer vision (CV) [23]. In practical applications of crowd counting, high accuracy and fast inference under limited computational resources are considered equally important [24], [25], [26]. Subsequent works reduce computation via knowledge distillation, shallow CNNs, and lightweight architectural refinements [24], [27]. In this work, we not only reduce computational cost through the deliberate design of network structures and convolutional modules, but also rethink the fundamental methodological choices.

Recent crowd counting methods can be grouped into three categories: CNN-based, transformer-based, and SSM-based approaches (see Fig. 2). CNN architectures often adopt dual-branch designs to enhance detail representation [28], [29], [30], [31], while hybrid models combine CNNs with transformers to improve feature interaction [32]. Although small-kernel convolutions reduce computation and preserve fine-grained head details, they weaken global modeling. To address this, we propose the DEmamba block, which couples CNNs with a Mamba-based backbone to retain global context at low cost.

B. Complex Background

When the pixel values in the background have similar values as that of the object, it can make learning hard. The primary

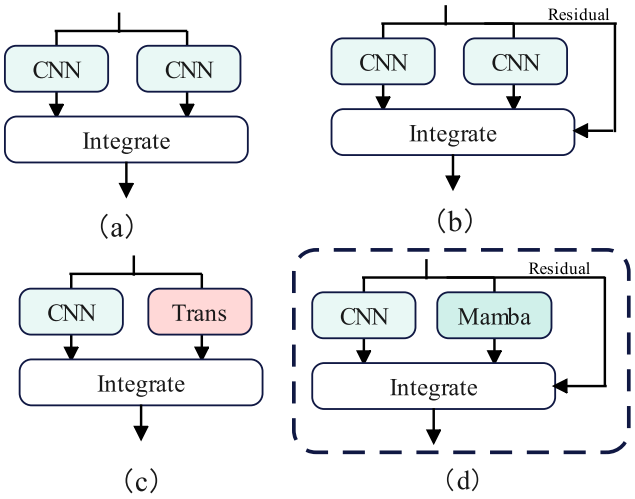


Fig. 2. Representative combinations of CNN, transformers, and SSM are illustrated. (a) Double CNN. (b) Double CNN (with residual). (c) CNN and transformers. (d) CNN and Mamba (with residual) (ours).

approach has been to incorporate various attention mechanisms to enhance the detection of key regions [33], [34], [35]. However, the attention-based methods mentioned above rely on CNNs, which often depend on fixed-size convolutional kernels to generate weights, making it challenging to balance local and global features. To address this limitation, we propose an adaptive algorithm that selects convolution kernel sizes dynamically, and we integrate it into standard CNN-based channel attention modules, forming the ACFM.

C. Variations in Scale and Density

Scale and density variations caused by perspective effects remain key challenges in crowd counting. One of the most classical solutions is multicolumn networks, such as MCNN [36]. Other methods attempt to generate more accurate density maps, such as setting Gaussian kernel sizes according to altitude information [27] or adding additional point annotations [37]. Some approaches rely on explicit geometric relationships, including inverse perspective mapping [12] and leveraging contextual cues from neighboring heads [38]. Further studies explore the relationship between sparse and dense scales to enhance scale awareness [39], while others classify regions into different density levels and then count them separately [40]. However, these approaches often suffer from high computational cost and limited adaptability, and they cannot be directly applied to point-based methods. In contrast, DACNet introduces a density-adaptive auxiliary point mechanism during training, which enhances robustness to density variations without relying on geometric information. It is also plug-and-play, allowing easy integration with other point-based methods without adding inference cost.

III. Methods

A. Overall Network Design

The architecture of the proposed model is illustrated in Fig. 3, and consists of four parts: the backbone, the neck, the denoising module, and the detection head. To effectively capture fine-grained features in densely populated regions, a lightweight detail-enhancement Mamba (DEmamba) module, is designed and integrated with standard convolutional layers to extract features from the original image. In the neck, a simplified version of the path aggregation feature pyramid network (PA-FPN) is employed, with emphasis placed on preserving the lower level, more detailed feature maps. Two of the low-level feature maps are further processed by the ACFM to suppress noise caused by variations in crowd density. Finally, the multiscale feature map obtained by concatenating the two refined feature maps is forwarded to the detection head, which not only generates the final predicted point coordinates P_{predict}, but also the coordinates of positive auxiliary points A_{pos} and negative auxiliary points A_{neg}. The predicted points and auxiliary points are matched to the ground truth points, and the losses for the point-based method are calculated as $\mathcal{L}_{\text{point}}$, with the auxiliary point losses given as $\mathcal{L}_{\text{APG}}^{\text{pos}}$ and $\mathcal{L}_{\text{APG}}^{\text{neg}}$. During the auxiliary point generation process, DA-APG dynamically adjusts the generation range of auxiliary points based on the density around the target point, reducing mismatches and capturing more long-range spatial information. This improves the model's performance in real-world scenarios with density variations.

B. Detail Enhancement Mamba Block

To efficiently extract fine-grained features in crowd images while reducing computational overhead, a common approach is to use small convolutional kernels. However, such kernels often struggle to capture global contextual information. In contrast, Mamba, an extension of the SSM, excels at global information modeling within a single layer but lacks the ability to effectively capture local detail. To address these complementary limitations, we propose the DEmamba module, which integrates both Mamba and convolutional mechanisms to achieve robust global context modeling alongside efficient local feature extraction. Specifically, as shown in Fig. 4, DEmamba comprises two branches: the lightweight counting Mamba (LCMamba) branch and the detail-enhancing convolution (DEC) branch.

The DEC branch adopts a cascaded 1×1 convolutional structure to enhance local details while significantly reducing parameters. The first convolution performs feature projection and dimensionality adjustment, while the second further refines local interactions and feature combinations. Given an input feature map with dimensions $H \times W \times C$, the output feature map is formulated as $F_o = \text{Conv}_{1\times1}(\text{Conv}_{1\times1}(F_i))$, where F_i is the input feature map and F_o is the output after two successive 1×1 convolutional layers.

The LCmamba branch leverages an improved Mamba architecture to capture global, long-range dependencies. Inspired by the work of Wang et al. [41], LCmamba is specifically adapted to address the limited feature richness commonly encountered in crowd counting tasks. Specifically, LCmamba incorporates additional residual connections to facilitate the reuse of shallow features such as edges and textures, which are frequently observed in crowd images. Furthermore, smaller 1×1 convolutional kernels are employed to extract

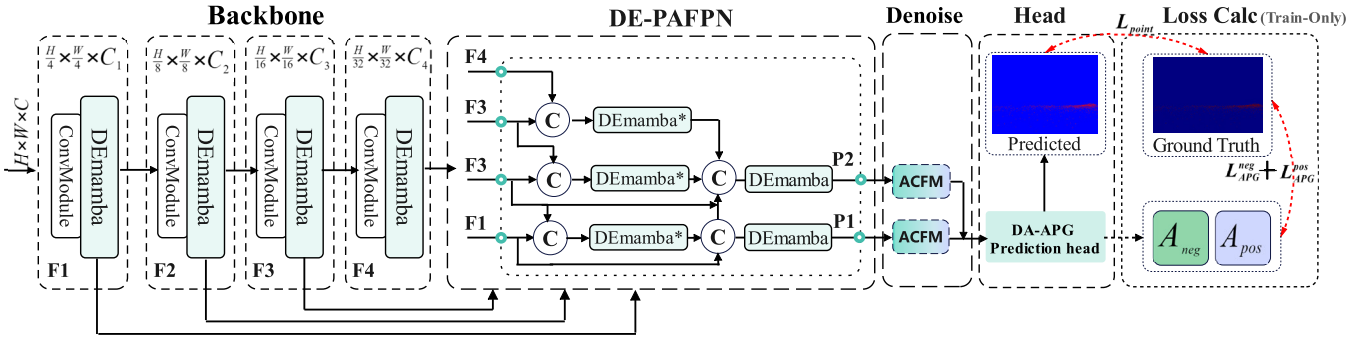


Fig. 3. DACNet can be roughly divided into four parts: the backbone composed of DEmamba, the PA-FPN (DE-PAFPN) composed of DEmamba, the denoising attention module composed of ACFM, and the prediction branch with DA-APG. The loss computation and auxiliary point generation are indicated with dashed lines, signifying that they are only involved during the training phase.

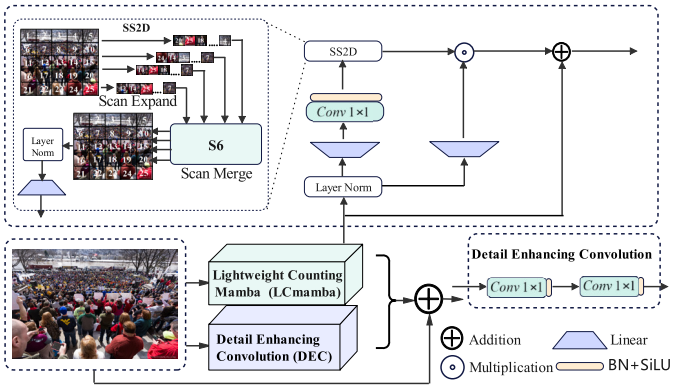


Fig. 4. Architecture of DEmamba. DEmamba combines two complementary components: Mamba and CNN. By incorporating a small-kernel design and an enhanced Mamba structure, DEmamba achieves efficient detail feature extraction while significantly reducing computational overhead.

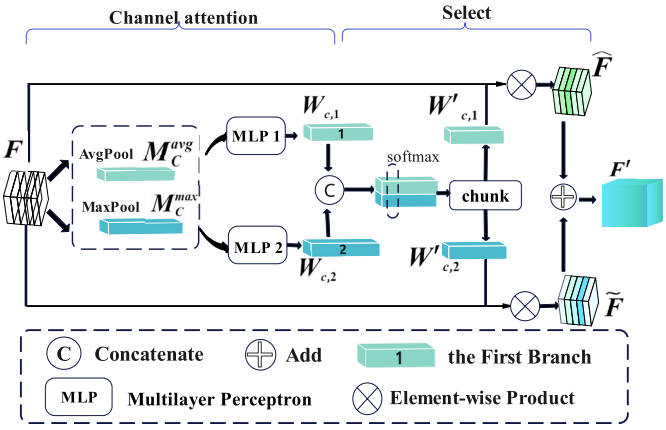


Fig. 5. Architecture of ACFM. The architecture is capable of selecting the advantages of different weights to balance local and global features, improving crowd density perception and reducing background noise interference.

finer-grained features, enhancing local detail representation while maintaining computational efficiency.

The input feature map F_i is first processed by layer normalization to ensure each channel has zero mean and unit variance, accelerating the training process. The normalized features are then passed through a fully connected layer followed by a 1×1 convolution to obtain an intermediate representation F_1. Global information is captured via the SS2D expansion operation, where we define $F_2 = \text{SS2D}(\text{Conv}_{1\times1}(\text{Linear}(\text{LayerNorm}(F_i))))$. Simultaneously, the normalized input is transformed again through a linear layer to obtain $F_3 = \text{Linear}(\text{LayerNorm}(F_i))$. The final output feature map is then computed as $F_{\text{out}} = F_2 \times F_3 + F_i$, using a residual connection that fuses the enhanced and original features. This structure effectively integrates global context and residual learning, enhancing the model's ability to detect small objects.

DEmamba also demonstrates advantages in terms of computational efficiency [42]. Given an input feature map $F \in \mathbb{R}^{C\times H\times W}$ and an output channel size of C', the computational complexity of the DEC branch (two 1×1 convolutional layers) can be expressed as $\text{FLOPs}_{\text{DEC}} = 2 \times C' \times C \times W \times H$. The total computational complexity of the DEmamba is the sum of the computational complexities of the DEC and LCmamba, and LCmamba has a relatively low computational cost. By incorporating residual connections and lightweight convolutional kernels, the DEmamba module significantly improves computational performance when processing crowd images, while also enhancing the accuracy of small object detection. These characteristics make it particularly well-suited for image analysis tasks in densely populated scenes.

C. Adaptive Channel Focus Module

The size of the convolution kernel affects the receptive field. Larger convolution kernels capture a wider range of features, making them better suited for extracting global features and handling larger scale objects. In contrast, smaller kernels are more effective at capturing local and detailed features, such as edges and textures [43]. To adapt to targets of varying scales, we proposed a channel selection structure, integrating it into different channel attention weights to form the ACFM. This multibranch structure fully leverages the advantages of different characteristic weights, producing unexpected results. We will discuss this in detail in Section IV-C. Specifically, we achieve adaptive channel focus by generating attention across multiple branches and balancing their weights. As shown in Fig. 5, only two branches are illustrated in this example, but the method can be easily extended to handle

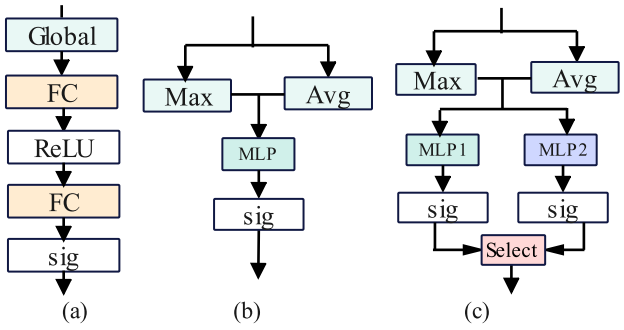


Fig. 6. Comparison between ACFM and other classical attention mechanisms. ACFM effectively leverages the advantage of multibranch weighting, while maintaining a computational complexity comparable to that of CBAM. (a) SE. (b) CBAM. (c) ACFM (ours).

multiple branches. In fact, based on the selection algorithm, it is also possible to fuse completely different channel attention mechanisms, combining the advantages of attention with varying characteristics.

1) Multibranch Attention Generation: Given a feature map $F \in \mathbb{R}^{C\times H\times W}$, to efficiently compute channel attention, we reduce the spatial dimensions of the input feature map. For spatial information aggregation, we adopt the method from CBAM [44], where both average pooling and max pooling are used to capture spatial information, resulting in two different feature maps denoted as M_{avg} and $M_{\max}$, corresponding to average-pooled and max-pooled features, respectively. These feature maps are then fed into two parallel multilayer perceptrons (MLPs) with different scales. The difference between the two MLPs lies in the varying spatial information they aggregate. The process of generating channel weights is as follows:

$$W_{c,i} = \sigma\left(\delta\left(\text{con}_i\left(\text{Avgpool}\left(F\right)\right) + \text{con}_i\left(\text{Maxpool}\left(F\right)\right)\right)\right).$$

Among them, $W_{c,i}$ is the channel attention weight output by the ith MLPs, where $W_{c,i} \in \mathbb{R}^{C\times 1}$. In this article, the sizes of the convolution kernels for the MLPs are 1×1 and 3×3, respectively.

2) Attention Selection: After obtaining the two branch weights, it is necessary to select the appropriate weights.. $W_{c,1}$ and $W_{c,2}$ are concatenated in the channel dimension, followed by the application of soft attention across channels to adaptively select information from different scales. The concatenated weights are split along the channel dimension using the chunk operation, forming two reweighted channel attention weights $W'_{c,1}$ and $W'_{c,2}$. Specifically, the selection in the channel direction follows the work of Li et al. [45], and is achieved through the softmax function as $a'_c = (e^{a_c}/(e^{a_c} + e^{b_c}))$ and $b'_c = (e^{b_c}/(e^{a_c} + e^{b_c}))$. Among them, a_c and b_c (where $c = 1, 2, \ldots, C$) represent the weights $W_{c,1}$ and $W_{c,2}$ on the cth channel, respectively. Similarly, a'_c and b'_c represent the weights $W'_{c,1}$ and $W'_{c,2}$ on the cth channel, respectively. Clearly, $a'_c + b'_c = 1$. Next, the selected attention weights $W'_{c,1}$ and $W'_{c,2}$ are multiplied element-wise with the input features F, forming two weighted features $\hat{F}$ and $\tilde{F}$. The two reweighted features are then summed element-wise to obtain the final feature F'. The algorithmic process for multibranch selection is depicted in the Supplementary Material.

We compare the architecture of ACFM with other representative channel attention mechanisms, such as SE [46] and CBAM [44]. Fig. 6 presents simplified schematics of each mechanism for a clearer structural comparison. In this comparison, only the channel attention part of the CBAM module is selected. For the channel attention part of CBAM, the main computational cost comes from the MLP branch, which contains two convolutional kernels of size 3×3. Given an input feature map with spatial dimensions $H \times W$ and channel dimension C, the FLOPs for this architecture are computed as $\text{FLOPs}_{\text{CBAM\_CA}} = 36 \times H \times W \times C^2$. In contrast, for ACFM, the main computational cost is increased by one additional MLP branch compared to CBAM's channel attention part, with the MLP branch containing two 1×1 convolutional kernels. For a 1×1 convolutional layer, the FLOPs can be formulated as $H \times W \times C^2$. Since ACFM introduces two additional parallel 1×1 convolutional branches, the total extra computational overhead is $\text{FLOPs}_{\text{Extra}} = 2 \times H \times W \times C^2$. Therefore, the total computational cost of ACFM is the sum of the computational cost of CBAM's channel attention and the additional overhead

$$\begin{aligned}\text{FLOPs}_{\text{ACFM}} &= \text{FLOPs}_{\text{Extra}} + \text{FLOPs}_{\text{CBAM\_CA}} \\ &= 38 \times H \times W \times C^2.\end{aligned}$$

It can be observed that the computational cost of ACFM is nearly the same as that of the channel attention part in CBAM, yet it performs significantly better in crowd images. We will further discuss this in detail in Section IV-C.

D. Density-Adaptive Auxiliary Point Guidance

Currently, the supervision process in traditional point-based methods [9] can be roughly summarized into the following three steps.

1) *Generate Point Proposals:* The model generates offset values and confidence scores, from which M point proposals (including the location of each point $\hat{\mathbf{p}}_i$ and the corresponding confidence $\hat{c}_i$) are obtained after transformation.
2) *Matching Between Ground Truth Points and Point Proposals:* The ground truth points are matched one-to-one with the generated point proposals using the Hungarian algorithm. The correctly matched point proposals are represented as $\hat{\mathcal{P}}_{\text{pos}} = \{\hat{\mathbf{p}}_{\xi(i)} \mid i \in \{1, \ldots, N\}\}$, and the unmatched point proposals are represented as $\hat{\mathcal{P}}_{\text{neg}} = \{\hat{\mathbf{p}}_{\xi(i)} \mid i \in \{N+1, \ldots, M\}\}$. Here, the predicted points that are closer to the ground truth and have higher confidence are more likely to be matched with the ground truth points.
3) *Compute Loss:* The final loss is obtained by calculating the classification loss L_{cls} and the localization loss L_{loc}

$$L_{\text{cls}} = -\frac{1}{M}\left(\sum_{i=1}^{N}\log\hat{c}_{\xi(i)} + \lambda_1\sum_{i=N+1}^{M}\log\left(1-\hat{c}_{\xi(i)}\right)\right) \quad (1)$$

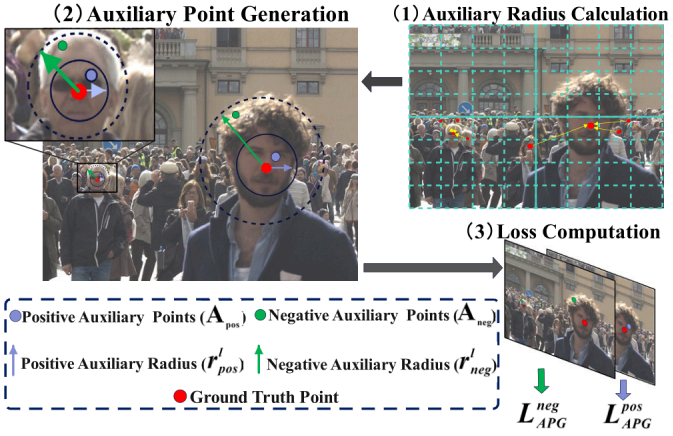


Fig. 7. DA-APG consists of three steps. In the process of (2) auxiliary point generation, DA-APG can adjust the auxiliary radius based on density variation.

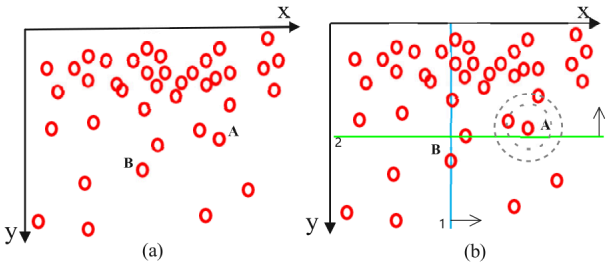


Fig. 8. Process of KD-Tree calculating the k-nearest neighbors.

$$L_{\text{loc}} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{p}_l - \hat{\mathbf{p}}_{\xi(i)}\|_2^2 \tag{2}$$

$$L_{\text{point}} = L_{\text{loc}} + \lambda_2 L_{\text{cls}}. \tag{3}$$

Inspired by the human brain, we hypothesize that generating additional points at varying distances around predicted locations can similarly enhance model performance. Building upon the work of Chen et al. [47], we propose the DA-APG method, which generates positive and negative auxiliary points at density-dependent distances to assist model training. These auxiliary points help guide the model to achieve accurate matching under varying density conditions. The operation of DA-APG is illustrated in Fig. 7 and consists of three main processes: auxiliary radius calculation, auxiliary point generation, and loss computation.

1) Auxiliary Radius Calculation: To adaptively adjust the auxiliary radius, we first estimate the local density ρ around each ground truth point, upon which the auxiliary radius is subsequently determined.

a) Local density estimation: Given a set of ground truth points $P = \{p_l = (x_l, y_l)\}_{l=1}^{N}$, where N is the total number of points, we calculate the local density ρ_l around each point by considering the k nearest neighbors. Specifically, we used a relatively resource-efficient method, KD-Tree (K-Dimensional Tree), to calculate the k-nearest neighbors. KD-Tree achieves nearest neighbor search through binary search, where the partitioning strategy alternates between each feature dimension. Fig. 8(a) shows the original distribution of head coordinates, where A is the query node and B is the root node. First, the x-axis position of node A is compared with node B, and it is partitioned into the right subtree. Next, the y-axis value of node A is compared with B's right child for further partitioning. This process is repeated recursively until the nearest node is found. After locating the nearest node, the local density ρ_l can be calculated as $\rho_l = (1/k)\sum_{i=1}^{k}(1/(\text{dist}(p_l, p_i) + \epsilon))$, where $\text{dist}(p_l, p_i)$ denotes the Euclidean distance between the point p_l and its ith nearest neighbor p_i, and ϵ is a small constant added to avoid division by zero.

b) Calculation of auxiliary radius: Based on the estimated local density ρ_l, we dynamically adjust the positive and negative auxiliary radii r^l_{pos} and r^l_{neg} for each real target point p_l, where $r^l_{\text{pos}} = (R_{\text{pos}}/(1+\alpha\rho_l))$ and $r^l_{\text{neg}} = (R_{\text{neg}}/(1+\beta\rho_l))$. Here, r^l_{pos} and r^l_{neg} denote the positive and negative auxiliary radius, respectively; R_{pos} and R_{neg} are the corresponding baseline radii; and α and β are the adjustment coefficients for positive and negative auxiliary points, which control the influence of density variations.

Our strategy ensures that in high-density regions, the positive auxiliary radius is smaller, while in low-density regions, the radius increases. In dense regions, the negative auxiliary radius increases, allowing negative auxiliary points to be generated further from the target point, thereby reducing interference with nearby positive points.

2) Auxiliary Point Generation: After calculating the auxiliary radius, auxiliary points are generated based on the varying radius. In dense areas, a smaller radius is used, while in sparse areas, a larger radius is applied to adapt to the changes caused by density variations. Both positive and negative auxiliary points are generated around each ground truth point p_l, with their generation adapting to local crowd density. In high-density areas, the generation radius is smaller to reflect the close proximity of individuals, while in low-density areas, the radius is larger to account for the sparser distribution. Specifically, for each ground truth point p_l, k_{pos} positive auxiliary points $A^i_{l,\text{pos}}$ and k_{neg} negative auxiliary points $A^j_{l,\text{neg}}$ are generated based on the auxiliary radius. The points are generated as follows:

$$A^i_{l,\text{pos}} = \left(x_l + R^{i,x}_{l,\text{pos}}, y_l + R^{i,y}_{l,\text{pos}}\right), \quad i = 1, 2, \ldots, k_{\text{pos}}$$
$$A^j_{l,\text{neg}} = \left(x_l + R^{j,x}_{l,\text{neg}}, y_l + R^{j,y}_{l,\text{neg}}\right), \quad j = 1, 2, \ldots, k_{\text{neg}}$$

where $R^{i,x}_{l,\text{pos}}$ and $R^{i,y}_{l,\text{pos}}$ are random values uniformly distributed within $[-r^l_{\text{pos}}, r^l_{\text{pos}}]$, and $R^{j,x}_{l,\text{neg}}$ and $R^{j,y}_{l,\text{neg}}$ are uniformly distributed within $[-r^l_{\text{neg}}, -r^l_{\text{pos}}] \cup [r^l_{\text{pos}}, r^l_{\text{neg}}]$. The radius r^l_{pos} and r^l_{neg} are adaptively determined based on the local crowd density, providing accurate supervision signals for both dense and sparse scenarios.

3) Loss Function Calculation: Auxiliary points provide additional supervision signals to address density variations, contributing to both positive and negative auxiliary point loss. The positive auxiliary point loss $L^{\text{pos}}_{\text{APG}}$ ensures predicted points are close to true target points with high confidence, while the negative auxiliary point loss $L^{\text{neg}}_{\text{APG}}$ encourages low confidence scores for points that are not true targets. The respective loss

TABLE I
DATASET STATISTICS

Dataset	Object	Training/Test	Count statistics		
			Min	Average	Max
VariDensity-CC	Crowd	320/80	362	776.3	1892
SHHA [36]	Crowd	300/182	33	501.4	3139
SHHB [36]	Crowd	400/316	9	123.6	578
JHU-Crowd++ [50]	Crowd	2772/1600	-	346	25791
Beijing_BRT [48]	Crowd	720/560	-	13.12	-
UCF_CC_50 [49]	Crowd	40/10	94	1280	4543
TRANCOS [51]	Car	824/422	-	37.6	-
SIA [52]	Soybean	101/25	35	-	296
SIB [52]	Soybean	106/28	46	-	295

TABLE II
COMPARISON OF DIFFERENT METHODS. THE BEST METHOD IS HIGHLIGHTED IN RED, WHILE THE SECOND-BEST METHOD IS HIGHLIGHTED IN BLUE

Method	Manner	Venue	SHHA		SHHB		UCF_CC_50		Beijing-BRT		JHU-Crowd++	
			MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
CSRNet [53]	Map-based	CVPR'18	68.2	115	10.6	16	266.1	397.5	1.7	2.4	-	-
P2PNet [9]	Point-based	ICCV'21	52.7	85.0	7.0	11.3	172.7	256.1	1.5	2.3	-	-
ChfL [54]	Map-based	CVPR'22	57.5	94.3	6.9	11	-	-	-	-	59.5	240.6
CLTR [38]	Point-based	ECCV'22	56.9	95.2	6.5	10.6	205.5	297.3	1.6	2.4	59.5	240.6
CrowdHat [55]	Detection-based	CVPR'23	51.2	81.9	5.7	9.4	175.7	246.1	-	-	52.3	211.8
PET [56]	Point-based	ICCV'23	49.3	78.8	6.2	9.7	168.7	236.7	1.3	2.2	58.5	238.0
CrowdDiff [57]	Map-based	CVPR'24	47.4	75.0	5.7	8.2	160.8	225.0	1.1	1.9	47.3	198.9
mPrompt [58]	Map-based	CVPR'24	52.5	88.9	5.8	9.6	-	-	-	-	-	-
MS2 [59]	Map-based	TNNLS'24	56.1	89.0	-	-	-	-	-	-	54.3	220.7
APGCC [47]	Point-based	ECCV'24	48.8	76.7	5.6	8.7	154.8	205.5	1.2	2.0	54.3	225.9
DACNet (Ours)	Point-based	-	47.4	74.9	5.7	8.5	156.8	199.1	1.0	1.8	52.5	211.4

functions are

$$L_{\text{APG}}^{\text{pos}} = \frac{1}{N}\frac{1}{k_{\text{pos}}}\sum_{l=1}^{N}\sum_{i=1}^{k_{\text{pos}}}\Big(-\log \hat{c}_{\text{pos}}(l,i) + \lambda_3 \|\mathbf{p}_l - \hat{\mathbf{p}}_{\text{pos}}(l,i)\|_2^2\Big)$$

$$L_{\text{APG}}^{\text{neg}} = \frac{1}{N}\frac{1}{k_{\text{neg}}}\sum_{l=1}^{N}\sum_{j=1}^{k_{\text{neg}}}\Big(\log\big(1-\hat{c}_{\text{neg}}(l,j)\big) + \lambda_4 \|\Delta_{\text{neg}}^{l,j}\|_2^2\Big).$$

In $L_{\text{APG}}^{\text{pos}}$, the classification term $\log \hat{c}_{\text{pos}}(l,i)$ ensures high confidence for positive points, and the regression term $\|\mathbf{p}_l - \hat{\mathbf{p}}_{\text{pos}}(l,i)\|^2$ penalizes deviations from true target points. For $L_{\text{APG}}^{\text{neg}}$, $\log(1-\hat{c}_{\text{neg}}(l,j))$ ensures low confidence for negative points, while $\|\Delta_{\text{neg}}^{l,j}\|_2^2$ reduces the offset of negative points. The total loss for DA-APG combines these auxiliary losses with the standard classification loss L_{cls} and localization loss L_{loc} from point-based methods

$$L_{\text{total}} = L_{\text{cls}} + \lambda_2 L_{\text{loc}} + \lambda_5 L_{\text{APG}}^{\text{pos}} + \lambda_6 L_{\text{APG}}^{\text{neg}}.$$

The DA-APG mechanism introduces an adaptive approach to generating auxiliary points based on the local density of target points, dynamically adjusting the auxiliary radius to improve the accuracy of point-based counting models. By incorporating both positive and negative auxiliary points into the training process, the model can better predict target locations and reduce false positives without increasing computational load, making it particularly effective in real-world scenarios.

IV. EXPERIMENT

A. Preparation of Experiment

1) Datasets: We collected a dataset specifically designed to capture variations in crowd density, called VariDensity-CC. The images were sourced from both our own on-site photography and publicly available footage. The annotation process was carried out independently by two professional annotators, followed by cross-validation and consistency checks to ensure high-quality and reliable labeling. we conduct extensive experiments, including six crowd datasets [36], [48], [49], [50], as well as the vehicle flow dataset TRANCOS [51] and the soybean dataset [52]. The description of the dataset is shown in Table I and the Supplementary Material.

2) Experimental Parameter: During the training process, each input image of irregular size was randomly cropped into four patches, each with a size of 128 × 128. In addition, each image was subjected to random horizontal flipping with a probability of 0.5 to enhance the data. The Adam optimizer was employed to optimize the model, with the batch size set to 4. A learning rate of 1×10^{-5} was established for convenient fine-tuning. Validation was performed on the validation set after every five training epochs. The weight parameters were defined as follows: $\lambda_1 = 0.5$, $\lambda_2 = 2 \times 10^{-4}$, $\lambda_3 = 1.5 \times 10^{-4}$, $\lambda_4 = 2 \times 10^{-4}$, $\lambda_5 = 0.3$, and $\lambda_6 = 0.2$. The baseline auxiliary radius (R_{pos} and R_{neg}) vary depending on the dataset. For most datasets, R_{pos} is set to 3, and R_{neg} is set to 10. For the traffic dataset TRANCOS, R_{pos} is set to 5, and R_{neg} is set to 12. The number of nearest neighbors k used for density estimation is set to 1, and the adjustment coefficients for the positive and negative auxiliary points, α and β, are set to 1×10^2. The number of positive and negative auxiliary points, K_{pos} and K_{neg}, is set to 2.

3) Evaluation Criteria: In the field of crowd counting, the performance of models is primarily reflected in two aspects: counting accuracy and localization accuracy. Specifically, we use MAE and mse to evaluate the counting performance of the model, and density normalized average precision (nAP) to evaluate its segmentation performance. The detailed definitions of these metrics are provided in the Supplementary Material.

B. Comparative Experiment

1) Counting Performance: We evaluated the performance of DACNet against other state-of-the-art methods using five popular crowd counting datasets [36], [48], [49], [50]. For the other models, the experimental settings and parameters provided by the official sources were adopted, and for models without publicly available code, results reported in their respective papers were used. All results were rounded to one decimal place, as shown in Table II. The experimental results demonstrate that our model exhibits superior overall performance. Specifically, on the benchmark SHHA dataset, our DACNet improved the MAE by 10.1% and the mse by 11.9% compared to the baseline model P2PNet. This significant improvement is attributed to DACNet's adaptability to density, its utilization of detailed features, and the reduction of noise, which will be further discussed in Section IV-D.

To evaluate the model's ability to handle density variations and generalize, we further compared DACNet with other point-based methods on the VariDensity-CC dataset, the TRANCOS traffic dataset [51], and the soybean dataset [52]. The test results are shown in Table III, where DACNet demonstrates a generally higher level of generalization. Thanks to the density-adaptive strategy in DA-APG, DACNet shows a

TABLE III
COMPARISON OF POINT-BASED METHODS ON VARIDENSITY-CC, TRANCOS, SIA, AND SIB DATASETS

Method	VariDensity-CC		TRANCOS		SIA		SIB	
	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE
P2PNet [9]	102.5	149.3	4.2	5.9	11.9	15.3	12.9	16.9
PET [56]	99.5	143.2	3.9	5.5	10.5	13.6	11.2	15.1
APGCC [47]	96.1	137.9	3.6	5.0	10.1	13.1	10.7	14.4
DACNet (Ours)	87.5	125.2	3.6	5.1	8.9	11.7	9.7	13.2

TABLE IV
PERFORMANCE OF DIFFERENT POINT-BASED MODELS ON EDGE COMPUTING DEVICES

Model	Parameters	FLOPs	FPS	PCP (W)	GPU (MiB)
P2PNet [9]	21.46M	114.68G	17.6	20.6	3642
PET [56]	51.46M	846.54G	2.98	29.6	23450
APGCC [47]	18.67M	98.75G	19.8	18.6	3172
DACNet (Ours)	**14.59M**	**72.3G**	**26.9**	**16.2**	**2481**

significant improvement on the high-density variation dataset, VariDensity-CC. On the TRANCOS and soybean dataset, the improvement of DACNet is relatively small. We hypothesize that this is due to the smaller scale and density variations in the dataset, which limits the potential benefits of the density-adaptive strategy. Nevertheless, DACNet still achieves the highest overall accuracy, which can be attributed to the enhanced detail extraction capability of the hybrid small-kernel structure in DEmamba and the multibranch advantages of ACFM. In fact, insufficient fine-grained features and severe background noise are common challenges in object counting tasks, which further underscore the superior performance of DACNet under such conditions.

2) Localization Performance: We compared our method with other point-based approaches on the SHHA and UCF_CC_50 datasets. As shown in Fig. 9(b), DACNet, which employs the density-adaptive method, achieved the highest counting and localization performance on the SHHA dataset, with an nAP of 0.72, representing an approximately 12.5% improvement over P2PNet. The yellow boxes on the right side of Fig. 9(a) further illustrate DACNet's ability to adapt to varying crowd densities, showing notable improvements in performance for both excessively large and extremely small targets. This improvement generalizes well to other object counting tasks—such as traffic flow and soybean counting. These density-adaptive strategies of DA-APG play a crucial role, resulting in a significant improvement in localization accuracy under varying densities.

On the other hand, we also observed discrepancies between the model's predictions and the ground truth annotations due to human labeling errors. As shown in the green boxes in Fig. 9(a), certain individuals on upper floors and distant vehicles were omitted by annotators, yet DACNet successfully detected them. We refer to such cases as "actually correct" detections, where the model outperforms the incomplete annotations. Moreover, in traffic flow counting scenarios, DACNet demonstrated increased sensitivity to small, distant objects that were also "actually correct." We hypothesize that this enhanced sensitivity is attributed to the superior density adaptation capability introduced by the DA-APG strategy.

3) Hardware Efficiency Evaluation: We compared the overall performance of other point-based methods on edge devices. Specifically, we evaluated various models in terms of the number of parameters, floating-point operations (FLOPs), power consumption, and memory usage. Power consumption was measured to assess the energy efficiency of each model on hardware, while memory usage was used to evaluate the resource utilization of the edge device during inference. During testing, an all-ones tensor with a size of (1, 3, 512, 512) was used as input, and the hardware's power consumption and memory usage were monitored using system-provided diagnostic tools. All models were deployed on the Nvidia Jetson AGX Orin edge computing platform, with inference accelerated using TensorRT (FP16 precision).

The test results are presented in Table IV. Thanks to the use of a simplified neck structure and the lightweight DEmamba module with small convolutional kernels, DACNet achieves a faster inference speed and is the only model with a frame rate (FPS) exceeding 25. As analyzed in Section III-B, this advantage is largely attributed to the backbone architecture: while other point-based models typically adopt VGG-16 as their backbone, the DEmamba-based backbone significantly reduces computational complexity compared to VGG-16. Moreover, because the DA-APG guiding strategy only incurs computational overhead during training and introduces no additional burden during deployment, DACNet also leads in both power efficiency and memory usage. In Section IV-E, we further validated the practical value of DACNet by applying it to real-world crowd counting tasks. The corresponding end-to-end real-time detection results have been made available through our GitHub repository.

C. Ablation Experiment

1) Effectiveness of DEmamba: To evaluate the effectiveness of the proposed DEmamba structure, we conducted ablation experiments with the ACFM module and the DA-APG strategy removed. Specifically, we examined several architectural configurations: a single CNN branch, a single Mamba branch, varying convolution kernel sizes within the Mamba branch, and the inclusion or exclusion of residual connections. The backbone network produced four feature maps, which were processed using a corresponding PAFPN structure at the neck. The experimental results are presented in Table V. When only a single Mamba or CNN branch was used, the network structure was overly simplistic, lacking either global or local feature representations, which resulted in decreased counting accuracy. In contrast, introducing residual connections and employing smaller convolution kernels led to significant performance improvements. This can be attributed to the ability of residual structures to retain fine-grained image details, while smaller kernels effectively compensate for the Mamba branch's limitations in local feature extraction.

In addition, we conducted ablation experiments on the neck structure, with results summarized in Table VI. The PAFPN architecture was adopted for the neck, and we varied the number of input feature maps. High-level feature maps contain more abstract and global information but tend to suffer from reduced localization accuracy. When more feature maps were input, the overall accuracy improved, primarily

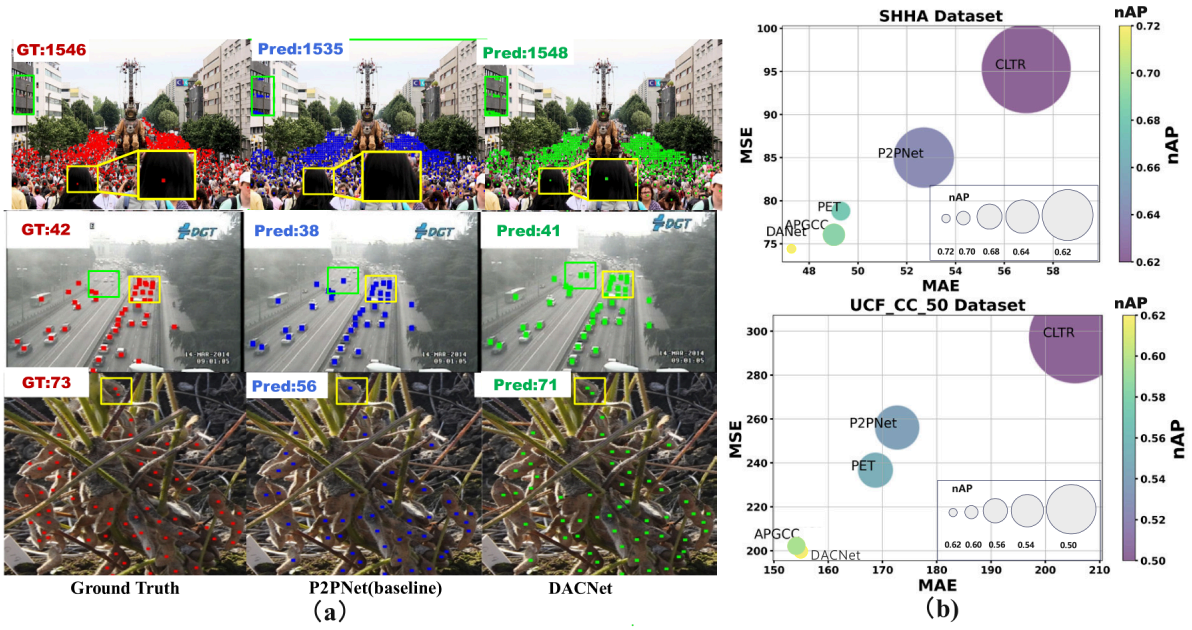


Fig. 9. Localization accuracy of DACNet. (a) Localization comparison with the baseline model, where the yellow boxes highlight DACNet's performance in both sparse and dense crowd scenarios. (b) Comparison of localization accuracy with other point-based methods.

TABLE V
EFFECTIVENESS OF DEMAMBA

Method	MAE	MSE	$nAP_{0.05}^{0.50}$
DEmamba (CNN only)	60.9	102.9	0.49
DEmamba (Mamba only)	57.4	94.9	0.54
DEmamba (Conv3&without residual)	55.6	89.4	0.58
DEmamba (Conv3&residual)	54.5	88.6	0.60
DEmamba (Conv1&without residual)	54.8	88.9	0.59
DEmamba (Conv1&residual)	53.5	88.3	0.62

TABLE VI
ABLATION STUDY ON THE NECK STRUCTURE

Input	Output	MAE	MSE	$nAP_{0.05}^{0.50}$
F1,F2,F3,F4	F1,F2,F3,F4	54.1	90.4	0.60
	F1,F2,F3	53.9 (+0.4%)	89.2 (+1.3%)	0.61 (+1.7%)
	F1,F2	53.5 (+1.1%)	88.3 (+2.3%)	0.62 (+3.3%)
	F1	53.7 (+0.7%)	88.9 (+1.7%)	0.62 (+3.3%)
F1,F2,F3	F1,F2,F3	54.5 (-0.7%)	91.2 (-0.9%)	0.58 (-3.3%)
	F1,F2	53.7 (+0.7%)	88.5 (+2.1%)	0.62 (+3.3%)
	F1	53.9 (+0.4%)	89.3 (+1.2%)	0.61 (+1.7%)
F1,F2	F1,F2	54.9 (-1.5%)	91.1 (-0.8%)	0.59 (-1.7%)
	F1	55.9 (-3.3%)	93.8 (-3.8%)	0.58 (-3.3%)

*All configurations are evaluated using the four-layer feature map setup as the baseline. Improvements are highlighted in red, while performance declines are shown in black.

TABLE VII
TEST RESULTS OF VARIOUS FEATURE PYRAMID METHODS ON THE SHHA DATASET

Method	MAE	MSE	$nAP_{0.05}^{0.50}$
ResNet-50 [60]	58.7	104.9	0.48
Swin-T [61]	56.3	95.4	0.52
Efficient VMamba [62]	55.1	91.7	0.58
MambaVision [63]	53.3	88.1	0.61
DEmamba (ours)	53.5	88.3	0.62

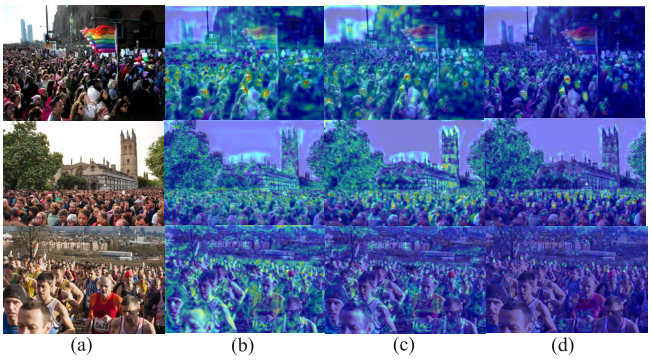
because high-level features provided a broader contextual understanding. However, when both the input and output included all four feature maps, the localization accuracy decreased due to the positional ambiguity introduced by high-level features. To balance semantic richness and localization precision, we ultimately selected a configuration that inputs four feature maps initially but outputs only the bottom two layers. This design preserves the advantages of high-level semantic information while minimizing the adverse effects of spatial ambiguity on localization accuracy.

To assess the impact of different backbones on the model, additional ablation experiments were performed with other common backbones. The neck structure remained as PAFPN, while only the backbone was changed, using four feature maps from each backbone. The results are shown in Table VII. ResNet-50, Swin-T, and Efficient VMamba represent typical convolutional, cision transformer (ViT), and Mamba structures, respectively. MambaVision combines all three. Single architectures inherently have limitations, while combining multiple structures facilitates the flow of both global and local information, resulting in higher accuracy. Although MambaVision achieved similar accuracy to DEmanbe, it incorporated additional ViT layers, increasing computational complexity. Therefore, DEmanbe was ultimately selected as the backbone network.

2) Effectiveness of ACFM: To validate the effectiveness of the ACFM module proposed in this article, we conducted experiments in other mainstream attention modules [44], [45], [46], [64]. The counting and localization results are presented in Table VIII. In addition, to verify the effectiveness of our proposed idea of selecting attention for different characteristics, we compared the performance of applying only the first branch weight and the second branch weight of ACFM separately. Surprisingly, this multibranch structure produced unexpected results. When using only the first branch weights, there was almost no improvement in accuracy. When using only the second branch weights, accuracy improved by about

TABLE VIII
TEST RESULTS OF ACFM AND OTHER ATTENTION MODULES ON THE SHHA DATASET

ID	MAE	MSE	$nAP_{0.05}^{0.50}$
DACNet*	53.5	88.3	0.62
DACNet* (With SE [46])	53.6 (-0.1%)	87.7 (+0.6%)	0.62 (+0%)
DACNet* (With CBAM [44])	53.4 (+0.1%)	87.9 (+0.4%)	0.63 (+1.6%)
DACNet* (With SK [45])	52.7 (+1.4%)	86.6 (+0.7%)	0.63 (+1.6%)
DACNet* (With CA [64])	51.7 (+3.3%)	84.6 (+3.7%)	0.64 (+3.2%)
DACNet* (With the First Branch)	53.4 (+0.1%)	87.9 (+0.4%)	0.63 (+1.6%)
DACNet* (With the Second Branch)	51.9 (+3.0%)	84.5 (+4.3%)	0.64 (+3.2%)
DACNet* (With ACFM)	49.1 (+8.2%)	77.6 (+12.8%)	0.66 (+6.4%)



(a) (b) (c) (d)

Fig. 10. Role of ACFM in noise reduction: (a) original image, (b) without attention, (c) CBAM, and (d) ACFM.

3%. However, when both branch weights were used together to form ACFM, accuracy increased by nearly 8%. This is because the first branch focuses only on local information, neglecting global information, while the second branch does the opposite, focusing solely on global information. ACFM, by using multi-branch weights and the weight selection algorithm, balances the perceived local and global information, leveraging the advantages of different weights, thus significantly improving accuracy.

To further validate the effectiveness of our proposed approach in balancing attention across different branches, we visualized the first-layer feature maps after the network's neck under three conditions: without an attention mechanism, with CBAM, and with ACFM. As shown in Fig. 10, the model's focus points are scattered and disorganized in the absence of an attention mechanism. After applying the CBAM module, the model's focus on the heads of the crowd improved, but a significant amount of noise was also captured. This is due to the fact that the CBAM module derives its weights solely from local detail features. In contrast, ACFM balances the attention weights between global and local features, effectively addressing the issues of noise and density variation.

3) Effectiveness of DA-APG: We compare the baseline method APG [47] with DA-APG and introduce both at the end of the model. The experimental results are shown in Table IX. Due to the use of the density-adaptive strategy, DA-APG effectively addresses the issues caused by density variation, reducing false positives in sparse scenarios and enhancing the guiding capability of auxiliary points in dense scenarios. Regarding the density adaptation of DA-APG, we will further discuss its impact on the prediction results in the visualization section of Section IV-D.

TABLE IX
COMPARISON OF APG AND DA-APG STRATEGIES ON SHHA DATASET

Strategy	MAE	MSE	$nAP_{0.05}^{0.50}$
DACNet*	49.1	77.6	0.66
DACNet*+APG [47]	48.9 (+0.4%)	77.0 (+0.7%)	0.68 (+3.0%)
DACNet*+DA-APG (ours)	47.4 (+3.4%)	74.9 (+3.5%)	0.72 (+9.0%)

TABLE X
EFFECT OF BASELINE AUXILIARY RADIUS (R_{pos}, R_{neg})

Baseline Auxiliary Radius (R_{pos}, R_{neg})	MAE	MSE
(1, 8)	49.4	76.9
(1, 10)	49.1	76.7
(1, 12)	50.0	78.3
(3, 8)	48.3	75.9
(3, 10)	47.4	74.9
(3, 12)	48.9	77.4

TABLE XI
EFFECT OF NEAREST NEIGHBORS k

The number of nearest neighbors k	MAE	MSE
1	47.4	74.9
2	47.2	74.6
3	47.1	74.4

TABLE XII
EFFECT OF ADJUSTMENT COEFFICIENTS α AND β

The adjustment coefficients α and β	MAE	MSE
(0.5, 0.5)	47.6	75.6
(1, 0.5)	47.5	75.3
(0.5, 1)	47.7	75.5
(1, 1)	47.4	74.9

4) Hyperparameters of DA-APG: We investigated the impact of different baseline auxiliary radius (R_{pos} and R_{neg}) and the number of neighboring points (k) on the model performance. Table X shows the effect of varying baseline auxiliary radius on model accuracy. The accuracy obtained by scheme (3, 10) is the highest. In fact, a range for negative auxiliary points that is too small may cause the model to be unable to effectively differentiate nontarget areas distant from the target point, mistakenly classifying some nontarget areas as target points. On the other hand, an overly large range for positive auxiliary points may cause the model to learn irrelevant features, particularly when the distance between target points is small and areas overlap. This, in turn, reduces the model's ability to distinguish between closely spaced target points.

In addition, the number of nearest neighbors k used for density estimation has minimal impact on model performance. As shown in Table XI, increasing k only slightly improves accuracy, while the computational cost doubles. Therefore, we selected $k = 1$ as the final parameter. Furthermore, the influence of the adjustment coefficients α and β is also minimal. In fact, the DA-APG strategy is capable of adapting to variations in these parameters during training. As shown in Table XII, different combinations of α and β result in comparable performance outcomes.

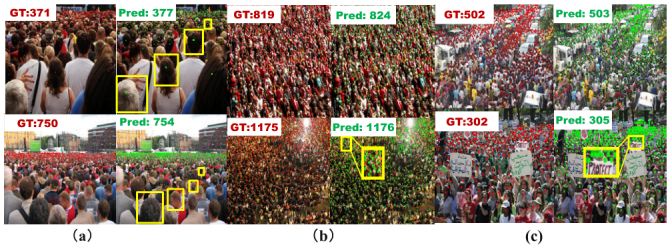


Fig. 11. Tests on typical images: (a) scale and density variations, (b) limited head features, and (c) severe background noise.

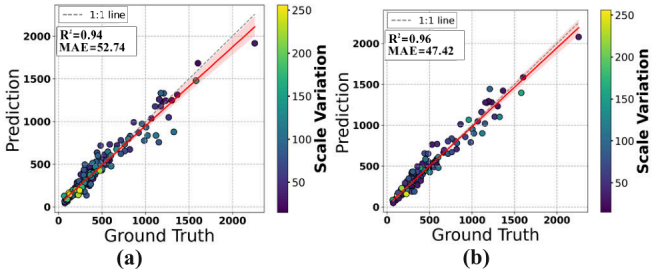


Fig. 12. Ability of the baseline model and DACNet to handle scale and density variation. (a) Baseline (P2PNet). (b) DACNet.

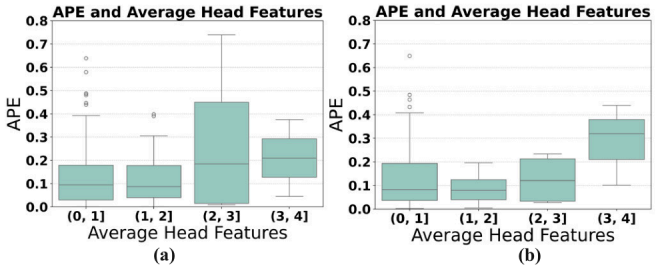


Fig. 13. Performance of the model under different head feature values (for intuitive representation, the average head features metric in the figure has been magnified by 10^3 times). (a) Baseline (P2PNet). (b) DACNet.

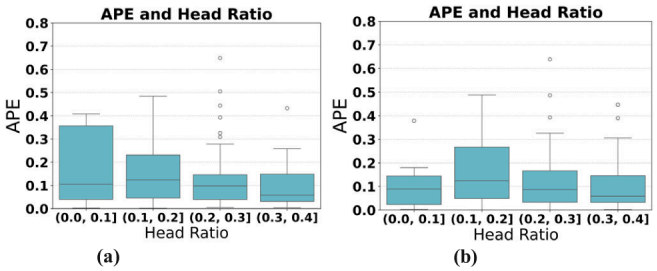


Fig. 14. Performance of the model under different levels of background noise. A higher head ratio indicates that the image contains more useful information and less background noise. (a) Baseline (P2PNet). (b) DACNet.

D. Visualization

Recall that the main motivation of this article is to address the issues of scale and density variations in crowd counting in real-world scenarios, as well as to mitigate the challenges of limited head features and complex background noise. To visually demonstrate the effectiveness of the proposed DACNet in these three aspects, we selected some typical images with scale and density variations, limited head features, and severe background noise to test the model, while also evaluating its generalization ability. As shown in Fig. 11, DACNet is able to effectively address the aforementioned issues. In addition, we discussed the improvements of DA-APG compared to the baseline model in addressing these three issues. We evaluate the relationships between scale and density variation, head features, background noise, and prediction accuracy. Experiments are conducted on the SHHA dataset, where the absolute percentage error (APE) is employed to quantify the prediction error for each image. APE reflects the percentage error between the predicted and actual head counts, with lower values indicating higher accuracy. The formula for APE is expressed as: $\text{APE} = |(y - \hat{y})/y| \times 100\%$, where y is the actual count and $\hat{y}$ is the predicted count.

1) Scale and Density Variation: Scale variation measures the differences in head sizes within an image. Fig. 12 show scatter plots of ground truth and predicted points for the two models (with the red line representing the fitting line). Fig. 12 shows that while both models degrade with larger scale and density variation, DACNet consistently maintains lower error and predictions closer to the 1:1 line.

2) Limited Features of the Head: We quantify image detail using canny edge detection: the ratio of detected edges to total pixels (edge ratio) reflects overall feature richness. We then define *Average Head Features* as Average Head Features = (Edge Ratio/Number of Heads), which normalizes the edge density per head. A higher value indicates richer per-head texture.

Fig. 13 illustrates the relationship between APE and Average Head Features. Due to the limited number of images in the $(2, 3]$ and $(3, 4]$ ranges, their reference value is not significant. However, it can be observed that DACNet reduces APE within the limited feature range of $[0, 1]$, demonstrating the effectiveness of DEmamba proposed in this article for extracting detailed features.

3) Background Noise: We estimate background noise using the head-to-image ratio, defined as the total estimated head area divided by the image area. For each head, we use KDTree to find the nearest neighbor distance, take half of it as the head radius, compute the corresponding area, and sum over all heads. A larger ratio implies more informative content (less background), while a smaller ratio indicates higher background noise.

As shown in Fig. 14, although DACNet's overall average APE reduction is not very significant, there is a relatively notable decrease in the $[0, 0.1]$ interval, where there are fewer heads and higher noise levels. This demonstrates the feasibility of using advanced features for noise reduction and highlights the effectiveness of the ACFM attention mechanism.

E. Model Deployment and Application

To evaluate the practical effectiveness of DACNet in real-world scenarios, we deployed the model on the edge computing device Nvidia Jetson AGX Orin for testing. As illustrated in Fig. 15, the system follows an end-to-end detection pipeline for edge deployment. In real-world tests, DACNet achieved a real-time processing speed of 26.9 FPS.

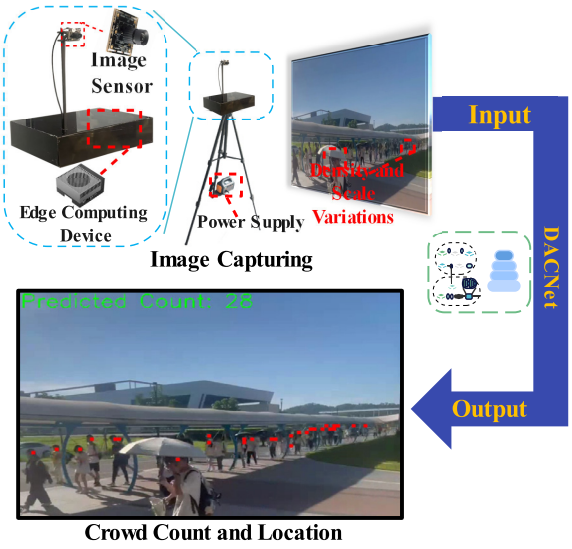


Fig. 15. Model deployment and real-world application. In practical scenarios, the camera height and viewing angle often cause substantial variations in head density and scale. DACNet enables end-to-end, real-time crowd detection and localization.

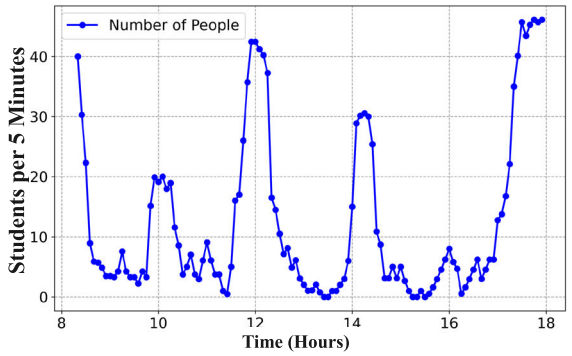


Fig. 16. Variation in crowd count along a specific section of the school over the course of a day.

We monitored pedestrian traffic along a specific walkway on campus from 8:00 A.M. to 6:00 P.M. This path connects classroom buildings with dormitory areas. The crowd count was measured in real time, and the average number of people was recorded every five minutes. The resulting trend is shown in Fig. 16.

V. Conclusion

In this article, we observe that in the practical application of crowd counting, there is an inevitable variation in crowd density due to the height and angle of the camera. However, existing end-to-endpoint-based methods fail to account for the aforementioned issues, resulting in limited accuracy. We propose DACNet, an end-to-end real-time crowd counting network designed to tackle the challenges posed by density variations in real-world scenarios. We propose a DA-APG strategy that effectively addresses the mismatch caused by density variation without adding extra computational overhead. Moreover, to address the challenges of insufficient detail in densely populated regions and background noise in real-world applications, we introduce the lightweight DEmamba module and the ACFM. DEmamba significantly reduces the complexity of the backbone network, while the multibranch structure of ACFM introduces minimal additional computational overhead. To advance research in this field, we publicly provide VariDensity-CC, a dedicated dataset specifically developed to reflect crowd density variations in real-world scenarios. Extensive experiments on nine datasets demonstrate that DACNet achieves the highest overall performance in both localization and counting, while maintaining real-time efficiency. DACNet is the only point-based model that achieves more than 25 FPS on edge devices. In future work, we plan to investigate the integration of DACNet with additional sensing modalities, including applications in smart city monitoring and security systems.

References

[1] A. Baihan, M. Amoon, T. Altameem, and M. Hashem, "Pioneering wearable sensor-driven health-monitoring system for contagious disease prevention with intelligent crowd-counting models," *IEEE Sensors J.*, vol. 25, no. 4, pp. 7403–7416, Feb. 2025.

[2] J.-H. Choi, J.-E. Kim, and K.-T. Kim, "Radar-based people counting under heterogeneous clutter environments," *IEEE Sensors J.*, vol. 24, no. 1, pp. 1028–1041, Jan. 2024.

[3] N. S. AlTakarli, "China's response to the COVID-19 outbreak: A model for epidemic preparedness and management," *Dubai Med. J.*, vol. 3, no. 2, pp. 44–49, May 2020.

[4] J. Weppner, P. Lukowicz, U. Blanke, and G. Tröster, "Participatory Bluetooth scans serving as urban crowd probes," *IEEE Sensors J.*, vol. 14, no. 12, pp. 4196–4206, Dec. 2014.

[5] L. Storrer et al., "Large-scale crowd size classification with a Wi-Fi-based passive radar," *IEEE Sensors J.*, vol. 24, no. 20, pp. 33560–33572, Oct. 2024.

[6] O. T. Ibrahim, W. Gomaa, and M. Youssef, "CrossCount: A deep learning system for device-free human counting using WiFi," *IEEE Sensors J.*, vol. 19, no. 21, pp. 9921–9928, Nov. 2019.

[7] 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 *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.

[8] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1324–1332.

[9] Q. Song et al., "Rethinking counting and localization in crowds: A purely point-based framework," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 3365–3374.

[10] M. A. Khan, H. Menouar, and R. Hamila, "Revisiting crowd counting: State-of-the-art, trends, and future perspectives," *Image Vis. Comput.*, vol. 129, Jan. 2023, Art. no. 104597.

[11] S. Basalamah, S. D. Khan, and H. Ullah, "Scale driven convolutional neural network model for people counting and localization in crowd scenes," *IEEE Access*, vol. 7, pp. 71576–71584, 2019.

[12] Y. Yuan, H. Guo, and J. Gao, "Distance-aware network for physical-world object distribution estimation and counting," *Pattern Recognit.*, vol. 157, Jan. 2025, Art. no. 110896.

[13] J. Cheng et al., "Scale, don't fine-tune: Guiding multimodal LLMs for efficient visual place recognition at test-time," 2025, *arXiv:2509.02129*.

[14] Y. Yuan, J. Zhao, C. Qiu, and W. Xi, "Estimating crowd density in an RF-based dynamic environment," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3837–3845, Oct. 2013.

[15] L. Lyu, C. Pang, and J. Wang, "Virtual-real syncretic crowd simulation method guided by building safety evacuation sign," *IEEE Sensors J.*, vol. 23, no. 16, pp. 18486–18499, Aug. 2023.

[16] X. Huang et al., "LEPS: A lightweight and effective single-stage detector for pothole segmentation," *IEEE Sensors J.*, vol. 24, no. 14, pp. 22045–22055, Jul. 2024.

[17] L. Zeng et al., "AI-based portable white blood cells classification and counting system in POCT," *IEEE Sensors J.*, vol. 24, no. 7, pp. 11057–11068, Apr. 2024.

[18] W.-C. Hu, L.-B. Chen, M.-H. Hsieh, and Y.-K. Ting, "A deep-learning-based fast counting methodology using density estimation for counting shrimp larvae," *IEEE Sensors J.*, vol. 23, no. 1, pp. 527–535, Jan. 2023.

[19] E. Hagenaars, A. Pandharipande, A. Murthy, and G. Leus, "Single-pixel thermopile infrared sensing for people counting," *IEEE Sensors J.*, vol. 21, no. 4, pp. 4866–4873, Feb. 2021.

[20] Y.-K. Cheng and R. Y. Chang, "Device-free indoor people counting using Wi-Fi channel state information for Internet of Things," in *Proc. GLOBECOM-IEEE Global Commun. Conf.*, Dec. 2017, pp. 1–6.

[21] A. Collaguazo, R. Estrada, I. Valeriano, and J. Tomala, "WiFi-crowd spy: A novel crowd-counting system," in *Proc. Int. Conf. Electr., Comput. Energy Technol. (ICECET)*, Jul. 2022, pp. 1–8.

[22] Z. Liu, R. Yuan, Y. Yuan, Y. Yang, and X. Guan, "A sensor-free crowd counting framework for indoor environments based on channel state information," *IEEE Sensors J.*, vol. 22, no. 6, pp. 6062–6071, Mar. 2022.

[23] W. Kong, H. Li, and F. Zhao, "Multiscale modality-similar learning guided weakly supervised RGB-T crowd counting," *IEEE Sensors J.*, vol. 24, no. 18, pp. 29121–29134, Sep. 2024.

[24] P. Wang, C. Gao, Y. Wang, H. Li, and Y. Gao, "MobileCount: An efficient encoder–decoder framework for real-time crowd counting," *Neurocomputing*, vol. 407, pp. 292–299, Sep. 2020.

[25] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[26] M. A. Khan, H. Menouar, and R. Hamila, "DroneNet: Crowd density estimation using self-ONNs for drones," in *Proc. IEEE 20th Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2023, pp. 455–460.

[27] M. A. Khan, H. Menouar, and R. Hamila, "LCDNet: A lightweight crowd density estimation model for real-time video surveillance," *J. Real-Time Image Process.*, vol. 20, no. 2, p. 29, Apr. 2023.

[28] B. Mu, F. Shao, Z. Xie, L. Xu, and Q. Jiang, "RGBT-Booster: Detail-boosted fusion network for RGB-thermal crowd counting with local contrastive learning," *IEEE Internet Things J.*, vol. 12, no. 11, pp. 18331–18349, Jun. 2025.

[29] J. Gao, L. Zhao, and X. Li, "NWPU-MOC: A benchmark for fine-grained multicategory object counting in aerial images," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5606614.

[30] C. Zheng, Z. Chen, X. Gao, and L. Lyu, "Dual backbone multi-attention hierarchical fusion and feature enhancement network for crowd counting," *IEEE Trans. Consum. Electron.*, vol. 71, no. 2, pp. 3279–3293, May 2025.

[31] J. Cheng et al., "MF-MOS: A motion-focused model for moving object segmentation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 12499–12505.

[32] Y. Liu, Y. Hu, G. Cao, and J. Wang, "SENet: Super-resolution enhancement network for crowd counting," *Pattern Recognit.*, vol. 162, Jun. 2025, Art. no. 111420.

[33] J. Gao, Q. Wang, and Y. Yuan, "SCAR: Spatial-/channel-wise attention regression networks for crowd counting," *Neurocomputing*, vol. 363, pp. 1–8, Oct. 2019.

[34] L. Rong and C. Li, "Coarse- and fine-grained attention network with background-aware loss for crowd density map estimation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2021, pp. 3675–3684.

[35] A. N. Alhawsawi, S. D. Khan, and F. U. Rehman, "Enhanced YOLOv8-based model with context enrichment module for crowd counting in complex drone imagery," *Remote Sens.*, vol. 16, no. 22, p. 4175, Nov. 2024.

[36] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 589–597.

[37] H. Zhao, Q. Wang, G. Zhan, W. Min, Y. Zou, and S. Cui, "Need only one more point (NOOMP): Perspective adaptation crowd counting in complex scenes," *IEEE Trans. Multimedia*, vol. 25, pp. 1414–1426, 2023.

[38] D. Liang, W. Xu, and X. Bai, "An end-to-end transformer model for crowd localization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, vol. 13661, 2022, pp. 38–54.

[39] S. D. Khan and S. Basalamah, "Sparse to dense scale prediction for crowd couting in high density crowds," *Arabian J. Sci. Eng.*, vol. 46, no. 4, pp. 3051–3065, Apr. 2021.

[40] A. N. Alhawsawi, S. D. Khan, and F. Ur Rehman, "Crowd counting in diverse environments using a deep routing mechanism informed by crowd density levels," *Information*, vol. 15, no. 5, p. 275, May 2024.

[41] Z. Wang, C. C. Li, H. Xu, X. Zhu, and H. Li, "Mamba YOLO: A simple baseline for object detection with state space model," in *Proc. AAAI Conf. Artif. Intell.*, 2025, vol. 39, no. 8, pp. 8205–8213.

[42] T. Chen et al., "MiM-ISTD: Mamba-in-Mamba for efficient infrared small target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, 2024, Art. no. 5007613.

[43] W. Luo, Y. Li, R. Urtasun, and R. S. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 4898–4906.

[44] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[45] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 510–519.

[46] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.

[47] I.-H. Chen, W. Chen, Y. Liu, M.-H. Yang, and S. Kuo, "Improving point-based crowd counting and localization based on auxiliary point guidance," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2024, pp. 428–444.

[48] X. Ding, Z. Lin, F. He, Y. Wang, and Y. Huang, "A deeply-recursive convolutional network for crowd counting," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1942–1946.

[49] H. Idrees, I. Saleemi, C. Seibert, and M. Shah, "Multi-source multi-scale counting in extremely dense crowd images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 2547–2554.

[50] V. A. Sindagi, R. Yasarla, and V. M. Patel, "JHU-CROWD++: Large-scale crowd counting dataset and a benchmark method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2594–2609, May 2022.

[51] R. Guerrero-Gómez-Olmedo, B. Torre-Jiménez, R. J. López-Sastre, S. Maldonado-Bascón, and D. Oñoro-Rubio, "Extremely overlapping vehicle counting," in *Proc. Iberian Conf. Pattern Recognit. Image Anal.*, 2015, pp. 423–431, 10.1007/978-3-319-19390-8_48.

[52] J. Zhao et al., "Improved field-based soybean seed counting and localization with feature level considered," *Plant Phenomics*, vol. 5, p. 26, Mar. 2023.

[53] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1091–1100.

[54] W. Shu, J. Wan, K. C. Tan, S. Kwong, and A. B. Chan, "Crowd counting in the frequency domain," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 19618–19627.

[55] S. Wu and F. Yang, "Boosting detection in crowd analysis via underutilized output features," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 15609–15618.

[56] C. Liu, H. Lu, Z. Cao, and T. Liu, "Point-query quadtree for crowd counting, localization, and more," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 1676–1685.

[57] Y. Ranasinghe, N. G. Nair, W. G. C. Bandara, and V. M. Patel, "CrowdDiff: Multi-hypothesis crowd density estimation using diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 12809–12819.

[58] M. Guo, L. Yuan, Z. Yan, B. Chen, Y. Wang, and Q. Ye, "Regressor-segmenter mutual prompt learning for crowd counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 28380–28389.

[59] H. Lin, X. Hong, Z. Ma, Y. Wang, and D. Meng, "Multidimensional measure matching for crowd counting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 36, no. 5, pp. 9112–9126, May 2025.

[60] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[61] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[62] X. Pei, T. Huang, and C. Xu, "EfficientVMamba: Atrous selective scan for light weight visual Mamba," in *Proc. AAAI Conf. Artif. Intell.*, 2025, vol. 39, no. 6, pp. 6443–6451.

[63] A. Hatamizadeh and J. Kautz, "MambaVision: A hybrid Mamba-transformer vision backbone," 2024, *arXiv:2407.08083*.

[64] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 13713–13722.