agriculture

MDPI

Article

M3DANet: A Lightweight Semi-Supervised Network and Embedded System for Bee Colony Counting

Xue Li [1,†], Mingzhen Ma [2,3,4,†], Ying Kong [5], Huijun Huang [6], Qian Li [6], Feng Liu [6], Zhenguo Liu [2,3,4,*] and Guangming Wang [7,*]

1 College of Information Science and Engineering, Shandong Agricultural University, Taian 271018, China; shelly_lixue@163.com
2 College of Animal Science and Technology, Shandong Agricultural University, Taian 271018, China; 2025120649@sdau.edu.cn
3 Key Laboratory of Efficient Utilization of Non-Grain Feed Resources (Co-Construction by Ministry and Province), Taian 271018, China
4 Shandong Provincial Key Laboratory of Animal Nutrition and Efficient Feeding, Taian 271018, China
5 College of Mechanical and Electronic Engineering, Shandong Agricultural University, Taian 271018, China; ky19837397860@163.com
6 Apiculture Institute of Jiangxi Province, Nanchang 330052, China; hhj_0929@163.com (H.H.); lq_07112026@163.com (Q.L.); liufeng801012@163.com (F.L.)
7 National Engineering Research Center of Agricultural Production Machinery and Equipment, Taian 271018, China
* Correspondence: zgliu@sdau.edu.cn (Z.L.); g.wang@sdau.edu.cn (G.W.)
† These authors contributed equally to this work.

Abstract

Accurate bee counting is important for colony monitoring, pollination assessment, and precision beekeeping, but manual counting and dense point annotation are labor-intensive. This study proposes M3DANet, a lightweight semi-supervised density regression network with a handheld edge deployment system for bee colony counting. A dataset containing 586 valid high-resolution images and 34,869 point annotations was constructed for training and evaluation. M3DANet uses the first seven stages of MobileNetV3-Large as the lightweight backbone and combines multi-scale context encoding, attention-guided low-level feature fusion, and teacher–student consistency learning with confidence masking and warm-up training. The 10%, 30%, and 50% labeled data settings refer to the proportions of labeled images in the training set, and the remaining training images are used as unlabeled data. Mean absolute error (MAE) and root mean square error (RMSE) are used as evaluation metrics. On the main dataset, M3DANet achieved MAE values of 9.937, 7.003, and 5.570 and RMSE values of 13.093, 9.387, and 7.620 under the 10%, 30%, and 50% settings, respectively, outperforming representative semi-supervised baselines. Under the fully supervised setting, it achieved an MAE of 5.201 and an RMSE of 6.989 with only 2.095 M parameters and 416.64 FPS, using 87.1% fewer parameters and running 17.7 times faster than CSRNet. Cross-species experiments confirmed its low-label generalization ability. Jetson Orin NX deployment achieved 65.75 ms/image inference latency and 10.44 FPS complete-pipeline throughput. These results show that M3DANet balances counting accuracy, annotation efficiency, generalization, and edge deployment practicality.

Keywords: bee counting; density map regression; semi-supervised counting; lightweight model; embedded deployment

Academic Editor: Aichen Wang

Received: 17 April 2026
Revised: 25 May 2026
Accepted: 8 June 2026
Published: 10 June 2026

1. Introduction

Bees and other pollinating insects play an essential role in maintaining ecosystem stability and ensuring global food security. Previous studies have shown that approximately one-third of global food production is directly or indirectly dependent on animal-mediated pollination, among which bees make a particularly important contribution [1]. Bee abundance is therefore a key quantitative indicator for evaluating pollination capacity, colony vitality, and ecological contribution. However, under the combined pressures of climate change, land use change, pesticide overuse, habitat degradation, and pathogen invasion, the health status of bee colonies has continued to decline worldwide, leading to increasing concern about the global pollination crisis [2]. To address this challenge, efficient and accurate monitoring of bee colonies is urgently needed. Traditional monitoring methods mainly rely on manual observation and expert experience, which are subjective, time-consuming, labor-intensive, and difficult to reproduce under large-scale field conditions. Therefore, accurate, efficient, and deployable bee counting has become an important technical problem in precision beekeeping and ecological monitoring.

With the rapid development of information technology and artificial intelligence, computer vision and deep learning have become important tools for bee colony monitoring and bee behavior analysis [3,4]. A recent systematic review on precision beekeeping emphasized that the Internet of Things, embedded sensing, and intelligent monitoring technologies are driving the transformation of modern beekeeping, and these technologies have been applied to hive status prediction, intrusion or predator detection, pest and disease monitoring, colony health assessment, and apiary management [5]. Among these applications, bee counting provides fundamental quantitative information for evaluating colony activity, population dynamics, pollination capacity, and management decisions. Benefiting from their strong feature representation capability, deep learning methods have been widely used in bee detection, classification, and behavior recognition [6–8]. In addition, visual counting and trajectory analysis techniques provide more reliable data support for colony vitality assessment and ecological monitoring [9–12]. Density-map-based counting methods estimate a continuous density distribution from point annotations and obtain the total number of targets by integrating the predicted density map during inference [13,14]. Owing to their robustness to occlusion, small targets, and scale variation, density regression methods have become a mainstream paradigm for dense scene counting [15,16]. Nevertheless, two major bottlenecks remain in practical bee counting applications. First, these methods usually require a large number of precise point annotations, resulting in high annotation costs when images contain numerous densely distributed bees. Second, high-accuracy models often have relatively high computational complexity, making them difficult to deploy directly on resource-constrained embedded devices. Therefore, reducing annotation dependence through semi-supervised learning while maintaining real-time counting capability through lightweight model design is crucial for promoting the practical application of bee colony monitoring systems [17,18].

In recent years, semi-supervised learning has been widely adopted in dense object counting because of its ability to exploit unlabeled data and reduce annotation requirements [19–21]. For example, in semi-supervised crowd counting, multi-task pseudo-label self-correction has been introduced to improve counting accuracy in dense scenes by constructing pseudo-label generation and self-correction mechanisms [22]. Multi-representation consistency learning has also been used to supervise unlabeled data by enforcing consistency constraints among different density representations, thereby enhancing robustness and generalization ability [23]. In addition, context-modeling-based semi-supervised methods combined with the Mean Teacher framework have been developed to guide pseudo-label generation for unlabeled samples and improve predictions in

both sparse and dense regions of complex scenes [24]. Related student–teacher approaches based on exponential moving average (EMA) have also been applied to fish recognition and detection tasks, where the teacher network generates pseudo-labels and the student network is trained with both labeled and unlabeled data [25]. These studies indicate that semi-supervised or weakly supervised learning provides an effective way to mine useful information from unlabeled samples and reduce the dependence on dense annotations [26,27]. However, most existing semi-supervised counting methods are designed for general dense targets such as crowds and vehicles, and they are not specifically optimized for bee counting scenarios. Bees are small, densely clustered, visually similar, and easily affected by occlusion, illumination changes, and complex hive backgrounds. As a result, pseudo-density maps generated by the teacher model may contain noise accumulation and counting bias. Therefore, bee counting tasks still require more reliable pseudo-supervision constraints and task-specific feature representation mechanisms.

Meanwhile, with the development of edge computing and the Internet of Things, deploying intelligent models on embedded devices has become an important trend for achieving real-time and low-power monitoring [28]. Lightweight backbone networks and model-compression strategies can significantly reduce inference latency and energy consumption on resource-constrained platforms. For example, MobileNetV3 provides a mobile-friendly architecture through neural architecture search and hardware-aware design, allowing models to maintain favorable performance on mobile and embedded devices [29]. In real-time object detection, the YOLO series has demonstrated a good balance between accuracy and speed for edge visual tasks [30]. Embedded platforms such as Jetson Nano and Jetson TX2 have also been widely used for low-latency vision applications, including counting and localization tasks [28]. In addition, early-exit and branch-pruning strategies can further improve embedded inference efficiency [28]. In beekeeping scenarios, recent edge device studies have explored tasks such as attention-integrated multi-scale predator detection for stingless bee protection and IoT-enabled honeybee monitoring for Varroa destructor detection using edge computing [31,32]. These studies demonstrate the potential of edge intelligence in precision agriculture and precision beekeeping. However, most existing edge intelligence studies in beekeeping focus on pest detection, predator detection, or hive status monitoring, whereas lightweight deployment for high-density bee counting remains insufficiently investigated.

In summary, existing studies have laid a solid foundation for bee visual monitoring, density-map-based counting, semi-supervised learning, and embedded deployment. However, several challenges remain unresolved. First, density-regression-based bee counting still depends heavily on dense point annotations, which limits its scalability in practical data collection. Second, bee images contain small, clustered, and visually similar targets under complex illumination and hive background conditions, requiring stronger fine-grained and multi-scale feature modeling. Third, practical beekeeping applications require not only accurate counting but also efficient inference on portable embedded devices. To address these gaps, this study proposes M3DANet, a lightweight semi-supervised density regression network for bee colony counting. The methodological novelty of M3DANet lies in its bee-counting-oriented integration of lightweight feature extraction, multi-scale context encoding, attention-guided low-level feature fusion, and confidence-masked teacher–student consistency learning. This design is intended to improve dense small-target representation, reduce dependence on fully labeled training data, and support real-time deployment on portable edge devices.

The main contributions of this study are summarized as follows:

1. A lightweight semi-supervised density regression network, named M3DANet, is proposed for bee colony counting. The network adopts MobileNetV3-Large as the

backbone and introduces a multi-scale context encoding (MSCE) module, in which atrous spatial pyramid pooling (ASPP) is followed by a multi-scale dilated convolution (MSDC) block to enhance contextual representation at different receptive field scales. In addition, an attention-guided low-level fusion (AGLF) module is designed to fuse low-level spatial details with high-level semantic features, thereby improving the representation of small and densely distributed bee targets while maintaining low model complexity and high inference efficiency.

2. A semi-supervised teacher–student learning framework with confidence region constraints is developed to reduce the dependence on dense manual annotations. The teacher network is updated using exponential moving average (EMA), and the student network is optimized with both labeled and unlabeled samples. Pixel-level density consistency and global count consistency are jointly imposed to constrain pseudo-supervision. In addition, a confidence mask based on high-response regions and a warm-up strategy are introduced to suppress unreliable pseudo-label interference and improve training stability under low labeled data ratios.
3. A dataset and portable edge-monitoring system tailored for field bee counting applications are established. The dataset contains 586 high-resolution images and 34,869 point annotations, covering representative variations in bee density, shooting distance, illumination, viewing angle, and hive background conditions. A label-consistent preprocessing workflow, including coordinate calibration, size normalization, and Gaussian density map generation, is constructed to ensure consistency across the training, testing, and deployment stages. Furthermore, a self-designed handheld bee-monitoring device is developed, and real-scene deployment tests are conducted to verify the practical feasibility of the proposed method in field beekeeping environments.

The remainder of this paper is organized as follows. Section 2 first presents the overall methodological framework of the proposed M3DANet-based bee colony counting system, and then describes the dataset construction and preprocessing workflow, the M3DANet network architecture, the feature extractor, the multi-scale context encoding module, the attention-guided low-level fusion module, the density regression head, the semi-supervised learning strategy, and the baseline implementation details. Section 3 presents the experimental setup and evaluation metrics, followed by comparisons with representative supervised and semi-supervised methods, structural and semi-supervised component ablation experiments, cross-species generalization experiments, sensitivity analysis of key hyperparameters, edge deployment verification, and failure case analysis. Section 4 discusses the main experimental findings; explains the contributions of the network architecture and semi-supervised learning strategy; analyzes the cross-species generalization results, deployment performance, and remaining limitations; and summarizes the practical significance of the proposed method. Finally, Section 5 concludes the paper and outlines future research directions.

2. Materials and Methods

2.1. Overall Methodological Framework

Figure 1 presents the overall methodological framework of the proposed M3DANet-based bee colony counting system. The framework starts from bee images, point labels, and unlabeled data. During data preprocessing, the input images are resized or cropped, point annotations are converted into Gaussian density maps, and a fixed data split is adopted for training, validation, and testing. The processed samples are then fed into M3DANet, which consists of a MobileNetV3-Large-based lightweight backbone, a multi-scale context encoding module, an attention-guided low-level fusion module, and a density regression

head. The predicted density map is further integrated to obtain the final bee count. During semi-supervised training, the teacher model is updated by exponential moving average, and confidence masks are used to constrain reliable regions for consistency learning. Finally, the trained model is deployed on a Jetson Orin NX-based handheld bee-counting device (NVIDIA Corporation, Santa Clara, CA, USA) for on-device inference, visualization, and real-time field counting.

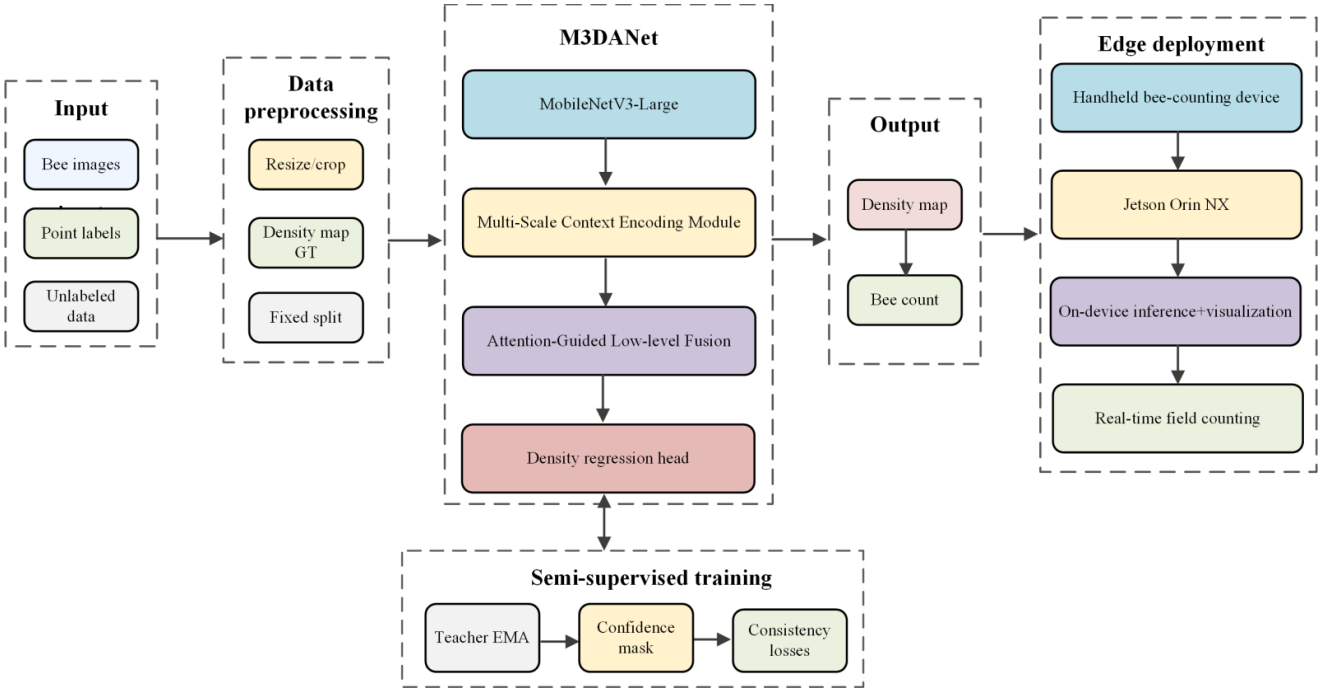

Figure 1. The overall methodological framework of the proposed M3DANet-based bee colony counting system.

2.2. Construction and Processing of the Dataset

The dataset used in this study was collected at the Science and Technology Innovation Park of Shandong Agricultural University in Shandong Province, China (117.1570° E, 36.1619° N), from June 2025 to March 2026. On-site images of bee colony activity were captured using a HUAWEI P50E smartphone (Huawei Technologies Co., Ltd., Shenzhen, China) under natural illumination. The original images had a resolution of 4096 × 3072 pixels. To reduce the effects of motion blur and unstable illumination, image acquisition was preferentially conducted during periods with stable weather and low wind speed. By adjusting the shooting angle, distance, and background conditions, bee colony images with different densities and scene characteristics were obtained, thereby enhancing the diversity and representativeness of the dataset. Manual image annotation was performed using VIA (VGG Image Annotator, version 2.0.12), and the generated CSV files recorded the image name and the x- and y-coordinates of each annotated point. The image-level ground-truth count was obtained by summing the annotated points in the corresponding CSV file. The final dataset contained 586 high-resolution images and 34,869 point annotations. Figure 2 presents visualization examples of representative samples from the bee counting dataset. For each sample, the preprocessed image, point annotation visualization, and corresponding generated density map are shown in sequence.

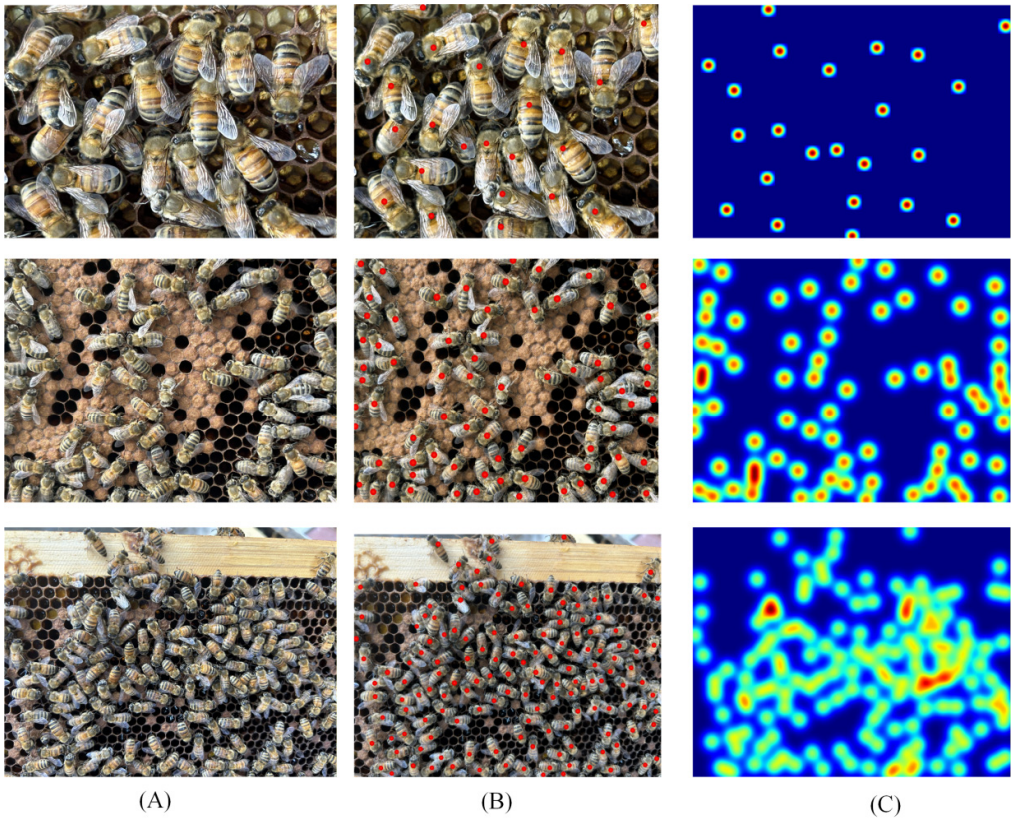

Figure 2. Visualization examples of the bee counting dataset. (**A**) Preprocessed images; (**B**) Point annotation visualizations, where the red dots indicate manually annotated bee locations; (**C**) Generated density maps, where warmer colors indicate higher local bee-density responses.

To adapt to the density map regression task, this study standardizes the preprocessing of the original image and point annotations to ensure strict alignment between image pixels and annotation coordinates. Firstly, parse the CSV annotation file, map the annotation points to the pixel coordinate system of the corresponding image, and correct or remove invalid points with inconsistent scales, repeated annotations, or exceeding the image boundary. Secondly, perform proportional normalization on the image size: scale the short sides to no less than 1024 pixels, and limit the long sides to no more than 4032 pixels; the point coordinates transform synchronously with the image scaling, forming standardized image annotation sample pairs. Finally, a geometric adaptive Gaussian kernel density map is generated based on normalized point labeling, with kernel width adaptively determined by local neighbor distance. To ensure counting consistency, the generated density map undergoes mass conservation calibration to ensure that the sum of all pixel values is equal to the actual number of bees in the image, that is:

$$\sum_{x,y} D(x,y) = N \tag{1}$$

where $D(x,y)$ represents the value of the density map at position (x,y), and N denotes the actual number of bees in the image.

The dataset was divided into a training set, a validation set, and a test set at a ratio of 7:2:1, resulting in 411 training images, 117 validation images, and 58 test images, which were used for model training, hyperparameter tuning, and final performance evaluation, respectively. All the labeled data ratio experiments used the same validation set and test set. Although point annotations were available for controlled evaluation, the semi-supervised settings were designed to simulate practical low-annotation scenarios, where

densely distributed bees make exhaustive point annotation time-consuming and labor-intensive. Therefore, semi-supervised training was performed only within the training set. For the 10%, 30%, and 50% labeled data settings, the corresponding proportions of training images were selected as labeled samples, while the remaining training images were treated as unlabeled samples. The labeled and unlabeled subsets were mutually exclusive and together covered the entire training set.

During the training phase, the standardized high-resolution images were randomly cropped into 768 × 768 patches, and data augmentation strategies such as horizontal flipping were applied to the labeled samples. The corresponding continuous density maps were cropped in the same spatial regions as the input images, and were then block-summed and downsampled according to the network output stride to generate supervised density maps with the same output resolution. In the experimental setup of this study, when the input image patch size was 768 × 768, the final supervised density map size was 192 × 192. This discretization process keeps the sum of the density map unchanged, ensuring consistency between density regression supervision and global counting supervision. For unlabeled samples, this study constructs weakly and strongly enhanced views from the same region for semi-supervised teacher–student consistency learning. Strong enhancements include operations such as brightness, contrast, color perturbation, and random Gaussian blurring [33]. In the validation and testing phases, random cropping is no longer performed, and only standardization processing is retained to ensure the stability and reproducibility of the evaluation results.

2.3. Design of M3DANet Network Architecture

For dense object counting in complex bee farm environments, this study proposes a semi-supervised density regression framework named M3DANet. The network structure is shown in Figure 3. The framework consists of a lightweight student network and a teacher branch based on exponential moving average (EMA): the former is used to predict density maps, while the latter provides more stable pseudo-supervision signals for unlabeled samples, thereby enhancing learning performance under limited annotation conditions.

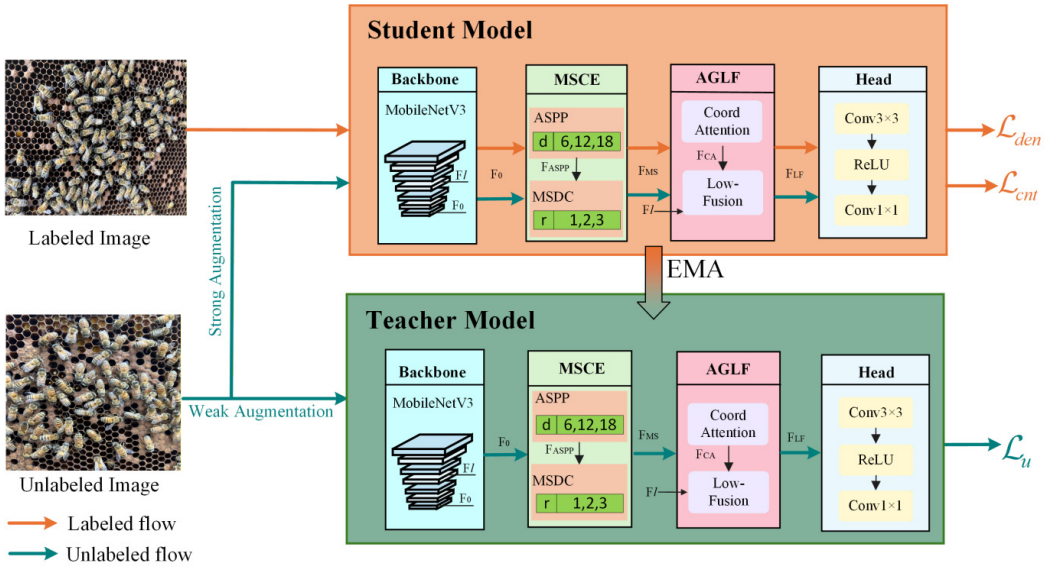

Figure 3. Network structure of M3DANet. MSCE consists of atrous spatial pyramid pooling (ASPP) followed by multi-scale dilated convolution (MSDC). AGLF denotes attention-guided low-level fusion, and EMA denotes exponential moving average. $\mathcal{L}_{den}$, $\mathcal{L}_{cnt}$, $\mathcal{L}_{u}$ denote the density regression loss, count loss, and unlabeled consistency loss, respectively.

In terms of network architecture, M3DANet first adopts the first seven stages of MobileNetV3-Large as the backbone, including inverted residual bottlenecks, depthwise separable convolutions, squeeze-and-excitation (SE) modules, and lightweight nonlinear activation functions [34]. This design reduces network depth, parameter size, and computational overhead while retaining sufficient shallow details and mid- to high-level semantic representation ability. The output of the fourth stage is used as a low-level feature map to preserve target edges, local textures, and spatial details, whereas the output of the seventh stage is used as a high-level feature map to provide stronger semantic and contextual representations. The high-level features are first mapped to 256 dimensions through a 3×3 convolution and are then sequentially fed into the atrous spatial pyramid pooling (ASPP) module and the multi-scale dilated convolution (MSDC) module to obtain a larger receptive field and richer multi-scale contextual information. Next, a lightweight attention module based on coordinate attention is utilized to enhance the high-level features by highlighting spatial responses related to target distribution. To improve the spatial detail restoration ability of the density map, the enhanced high-level features are upsampled to the resolution of the low-level features, concatenated with the low-level features after a 1×1 projection, and then fused and refined through two 3×3 convolution layers. Finally, a single-channel density map is output through a lightweight density regression head, and the final counting result is obtained by integrating the density map.

During training, the labeled samples are jointly optimized using the density regression loss and count constraint, whereas the unlabeled samples are constrained through density consistency and count consistency between weakly and strongly augmented views, thereby enhancing the model's generalization ability under limited annotation conditions.

2.3.1. Feature Extractor

To balance feature representation and computational efficiency, this study uses the first seven feature blocks of MobileNetV3-Large as a lightweight feature extractor [29]. As shown in Figure 4, the feature extractor generates two-level features from the input image ($3 \times H \times W$). The first stage consists of a 3×3 standard convolution, batch normalization, and hard-swish activation, and performs initial downsampling with a stride of 2 to extract basic color, edge, and texture responses. The second stage uses a 3×3 depthwise separable residual block with a stride of 1 to refine shallow features while maintaining the spatial resolution. The third stage reduces the feature resolution to $H/4 \times W/4$ through pointwise expansion, stride-2 depthwise convolution, and pointwise projection. The fourth stage further enhances low-level structural representations at the $H/4 \times W/4$ resolution, and its output is selected as the low-level feature F_l to preserve bee boundaries, fine-grained textures, and local spatial details. The fifth stage introduces a 5×5 depthwise convolution and SE attention, and reduces the feature resolution to $H/8 \times W/8$ to capture a wider range of local context. The sixth stage further aggregates mid-level semantic responses and suppresses redundant background information at the $H/8 \times W/8$ resolution. The seventh stage continues to use 5×5 depthwise convolution and SE calibration to produce the high-level feature F_h, which provides stronger semantic and contextual representations for subsequent density regression. After backbone extraction, the high-level feature F_h is further processed by a 3×3 convolution followed by ReLU to obtain F_0 with a size of $256 \times H/8 \times W/8$, while the low-level feature F_l is retained for later fusion to enhance spatial detail recovery in density prediction.

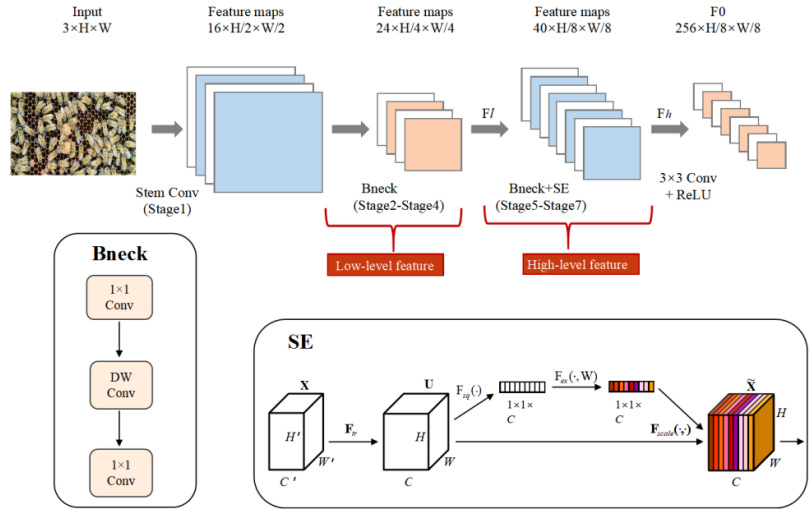

Figure 4. The structure of the feature extractor.

2.3.2. Multi-Scale Context Encoding Module

In medium- to high-density bee colony scenes, local neighborhood textures and larger range flow patterns are equally important for counting. To this end, this study introduces a multi-scale context encoding (MSCE) module consisting of atrous spatial pyramid pooling (ASPP) and multi-scale dilated convolution blocks (MSDC) [35,36]. To this end, M3DANet has designed two modules in series on the output feature Fo of the backbone network, namely ASPP and multi-scale dilated convolution block (MSDC) [35,36], for explicitly modeling multi-scale contextual information and fine-grained aggregation of local structures. The overall structure is shown in Figure 5.

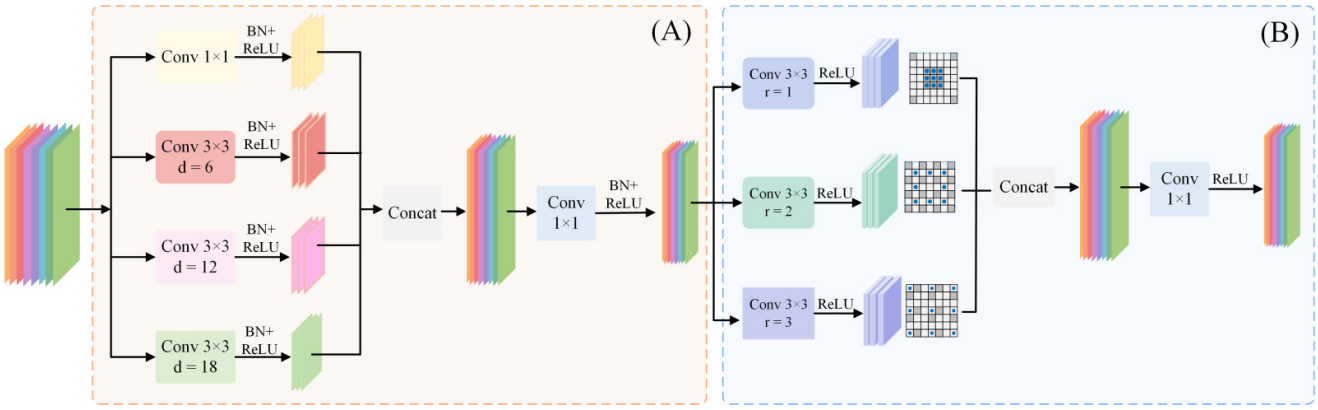

Figure 5. Structure of MSCE: (**A**) ASPP; (**B**) MSDC.

Firstly, the ASPP module is mainly responsible for encoding global and large-scale contexts. As shown on the left side of Figure 5, ASPP consists of four parallel branches: one branch is a 1 × 1 convolution used to preserve the original local response; the other three have expansion rates of d1 = 6, d2 = 12, and d3 = 18, respectively. The 3 × 3 dilated convolution with a dilation rate of 18 expands the receptive field without reducing spatial resolution, covering the overall motion pattern from a small gathering area near the entrance of the bee colony to a larger range. The output features of the four branches are concatenated in the channel dimension, and then linearly fused through a 1 × 1 convolution to obtain a unified multi-scale contextual feature:

$$F_{ASPP} = Conv_{1\times1}\left(Cat\left[Conv_{1\times1}(F_0), Conv_{3\times3}^{d_1}(F_0), Conv_{3\times3}^{d_2}(F_0), Conv_{3\times3}^{d_3}(F_0)\right]\right) \quad (2)$$

In Equation (2), F_{ASPP} represents the multi-scale contextual features output by the ASPP module, which are obtained by concatenating and fusing a 1 × 1 convolution branch and three 3 × 3 dilated convolution branches with different dilation rates.

The coverage of receptive fields corresponding to different expansion rates can be illustrated in Figure 6, where larger expansion rates can perceive the distribution patterns of bee colonies over a larger range.

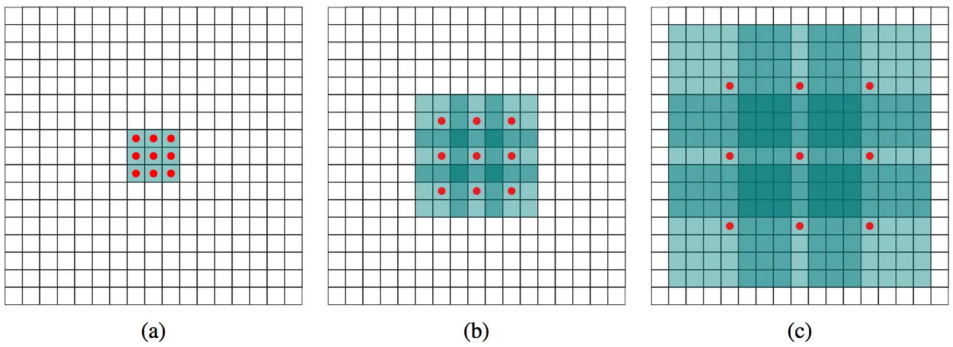

Figure 6. Receptive fields corresponding to different dilation rates: (**a**) dilation rate = 1; (**b**) dilation rate = 2; (**c**) dilation rate = 3. The red dots indicate the sampling positions of the convolution kernel, and the teal shaded regions indicate the receptive-field coverage under different dilation rates.

On the basis of ASPP output F_{ASPP}, to further enhance the modeling ability of local structures and high-density region details, M3DANet introduces a multi-scale dilated convolution block MSDC. Compared with ASPP which focuses on large receptive fields, MSDC focuses on finely characterizing the local geometric shape and occlusion relationship of bee colonies within a smaller expansion rate range [35,36]. As shown on the right side of Figure 5, MSDC applies three parallel 3 × 3 dilated convolution branches to the same input feature, with dilation rates of r1 = 1, r2 = 2, and r3 = 3, respectively. Then, they are concatenated in the channel dimension and aggregated and compressed through 1 × 1 convolution:

$$F_{MS} = Conv_{1\times1}\left(Cat\left[Conv^{r_1}_{3\times3}(F_{ASPP}), Conv^{r_2}_{3\times3}(F_{ASPP}), Conv^{r_3}_{3\times3}(F_{ASPP})\right]\right) \quad (3)$$

In Equation (3), F_{MS} denotes the multi-scale local structural feature obtained after MSDC processing. It is generated by applying three parallel 3 × 3 dilated convolution branches with different dilation rates to F_{ASPP} for fine-grained local structure modeling, followed by channel-wise concatenation and 1 × 1 convolutional aggregation.

2.3.3. Attention-Guided Low-Level Fusion

Given the MSCE features F_{MS} the network is further followed by an attention-guided low-level fusion (AGLF) module, composed of coordinate attention (CA) and low-level feature fusion [37,38], aiming to selectively enhance informative semantic responses while recovering fine spatial details. Figure 7 shows the structure of AGLF.

Coordinate attention first performs global average pooling along the vertical and horizontal directions, thereby explicitly encoding contextual information along the row and column directions and obtaining two direction-aware feature descriptors. After concatenation along the spatial dimension, the aggregated representation is passed through a 1 × 1 convolution, batch normalization, and ReLU for channel compression and feature interaction, and is then split into two attention maps, A_h and A_w, corresponding to the

height and width directions, respectively. Finally, coordinate attention recalibrates the input feature map through element-wise multiplication:

$$F_{CA}(c, y, x) = F_{MS}(c, y, x) \cdot A_h(c, y, 1) \cdot A_w(c, 1, x) \tag{4}$$

In Equation (4), F_{CA} represents the high-level semantic feature recalibrated by coordinate attention, which incorporates positional dependency information along both the horizontal and vertical directions.

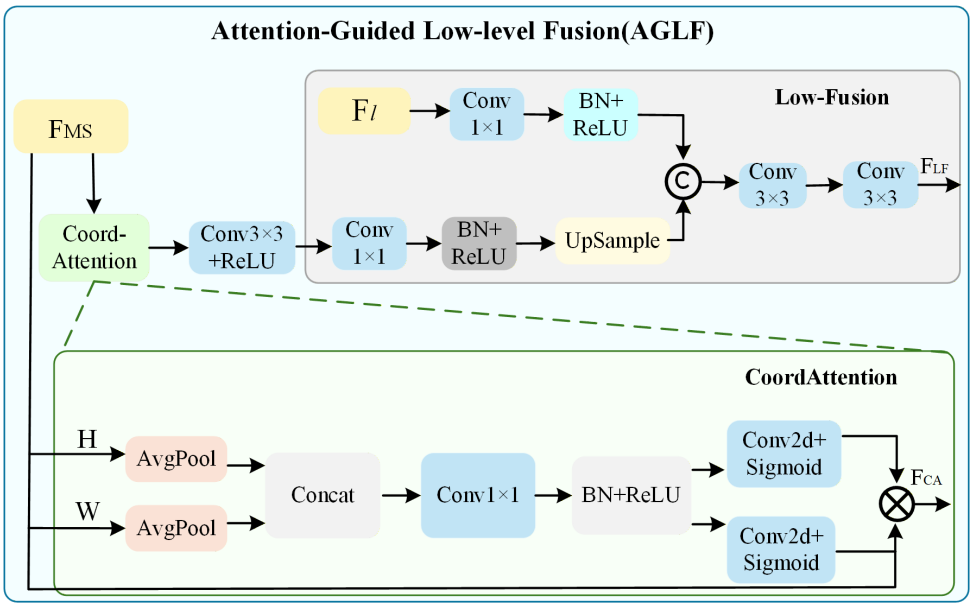

Figure 7. Structure of AGLF.

This operation enhances spatially informative regions related to bee distribution while suppressing irrelevant background responses.

After coordinate attention, the enhanced high-level feature is first refined by a 3×3 convolution with ReLU, and then fused with the low-level feature extracted from an earlier backbone stage to recover fine-grained boundary and texture information. Specifically, the high-level feature is projected by a 1×1 convolution followed by batch normalization and ReLU, and then upsampled to the spatial resolution of the low-level feature. Meanwhile, the low-level feature is also projected to a unified channel dimension through another 1×1 convolution with batch normalization and ReLU. The two feature maps are then concatenated along the channel dimension and refined by two consecutive 3×3 convolution layers, producing the final fused feature F_{LF}:

$$\begin{aligned} \tilde{F}_h &= \mathrm{Up}(\phi_h(F_{CA})), \quad \tilde{F}_l = \phi_l(F_l) \\ F_{LF} &= \psi\left(\mathrm{Concat}\left(\tilde{F}_h, \tilde{F}_l\right)\right) \end{aligned} \tag{5}$$

where $\phi_h(\cdot)$ and $\phi_l(\cdot)$ denote 1×1 convolutional projections followed by BN and ReLU, and $\psi(\cdot)$ denotes the feature refinement function implemented by two stacked 3×3 convolutions. F_{LF} denotes the feature obtained after fusing low-level detailed information, which is used for the subsequent density regression.

Overall, the AGLF module can be summarized as:

$$F_{LF} = \mathrm{Fuse}(\mathrm{CA}(F_{MS}), F_l) \tag{6}$$

Among them, CA (·) and Fuse (·) respectively represent coordinate attention and low-level feature fusion operator.

2.3.4. Density Regression Head and Count Estimation

Based on the AGLF feature F_{LF}, M3DANet employs a lightweight convolutional regression head to generate the final density map. Specifically, the regression head consists of a 3×3 convolution ($128 \rightarrow 64$), followed by a ReLU activation, and a 1×1 convolution ($64 \rightarrow 1$). A final ReLU operation is applied to ensure non-negative density output. The overall mapping can be written as:

$$\hat{D} = \mathrm{ReLU}(\phi_{head}(F_{LF})) \tag{7}$$

Among them, ϕ_{head} denotes the regression head composed of two convolutional layers with an intermediate ReLU activation. Since the proposed network adopts low-level feature fusion, the predicted density map is generated at a spatial resolution of $H/4 \times W/4$ under the current setting.

After obtaining the predicted density map $\hat{D}$, the estimated number of bees in the entire image can be obtained by summing up all pixel positions:

$$\hat{C} = \sum_{i,j} \hat{D}(1, i, j) \tag{8}$$

Among them $\hat{C}$ denotes the predicted count. Since the Gaussian density map used during training satisfies the mass conservation constraint, i.e., the integral of the density map is equal to the number of annotated points, the above summation remains naturally consistent with manual point-based counting during inference. At the same time, this operation introduces very little computational overhead, making it suitable for real-time deployment on embedded platforms.

2.4. Semi-Supervised Learning Strategy and Loss Functions

In this work, a teacher–student framework is adopted for semi-supervised learning. The student network is trained on a limited number of labeled samples and further constrained by consistency learning on unlabeled samples, whereas the teacher network is updated via exponential moving average and provides more stable pseudo-supervision signals for unlabeled data. Within this framework, the supervised component is used to optimize density map regression and overall counting accuracy, while the consistency component helps the model learn more stable density representations from unlabeled samples.

2.4.1. Supervised Loss of Labeled Samples

For labeled samples, the model is jointly optimized by a pixel-level density regression loss and an image-level count loss. Let a batch contain B labeled samples. For the b-th sample, the ground-truth density map is denoted by D_b^*, and the corresponding ground-truth bee count is denoted by C_b^*. The predicted density map is denoted by $\hat{D}_b$. The predicted count can then be computed by summing the values over all spatial locations of the predicted density map, as follows:

$$\hat{C}_b = \sum_{i,j} \hat{D}_b(i, j) \tag{9}$$

The ground-truth density map $\hat{D}_b$ is generated from the manual point annotations described using an adaptive Gaussian kernel and is further calibrated by mass conservation. The ground-truth count $\hat{C}_b$ is the total number of valid manually annotated points in the

corresponding image, which is also equal to the integral or discrete summation of $\hat{D}_b$ over the whole image.

The density regression term is defined as the mean squared error between the predicted and ground-truth density maps:

$$\mathcal{L}_{den} = \frac{1}{B}\sum_{b=1}^{B}\left\|\hat{D}_b - D_b^*\right\|^2 \tag{10}$$

To further constrain the global counting accuracy, a normalized count loss is introduced:

$$\mathcal{L}_{cnt} = \frac{1}{B}\sum_{b=1}^{B}\omega_b\frac{|\hat{C}_b - C_b^*|}{C_b^* + \epsilon_c} \tag{11}$$

where ϵ_c is a small constant for numerical stability, and ω_b denotes the weight assigned to high-count samples, defined as

$$\omega_b = \min\left((C_b^* + 1)^\beta, \gamma\right) \tag{12}$$

Accordingly, the supervised loss for labeled samples is formulated as

$$\mathcal{L}_l = \lambda_{den}\mathcal{L}_{den} + \lambda_{cnt}\mathcal{L}_{cnt} \tag{13}$$

In Equation (13), λ_{den} and λ_{cnt} denote the weighting coefficients of the density regression loss and the count loss, respectively. The density regression loss and the count loss have different numerical scales because the former is computed from pixel-level density map regression, whereas the latter is computed from image-level count errors. Therefore, λ_{den} and λ_{cnt} are introduced to balance the magnitudes of the two loss terms during optimization. In the final experimental setting, $\lambda_{den} = 20$, $\lambda_{cnt} = 0.01$, $\epsilon_c = 1.0$, $\beta = 0.1$, and $\gamma = 4.0$. This loss formulation takes density regression as the main optimization objective, while introducing count supervision to improve global counting consistency. In addition, the normalized count error and the mild reweighting of high-count samples help stabilize the training process across scenes with different crowding levels.

2.4.2. Consistency Loss of Unlabeled Samples

When labeled data is limited, relying solely on them can lead to overfitting. To fully utilize the information of unlabeled images, a teacher–student framework was constructed: the student network S is used for gradient updates, and the teacher network shares the same structure, updating student parameters through exponential moving average (EMA) [19]:

$$\theta_t \leftarrow \rho\theta_t + (1-\rho)\theta_s \tag{14}$$

Among them, θ_s and θ_t are the parameters for students and teachers respectively, and ρ is the EMA attenuation coefficient, which is set to 0.999 in this article.

For each unlabeled sample, the teacher network receives a weakly augmented view, while the student network receives a strongly augmented view. Let the corresponding density predictions be denoted by $D_u^{(t)}$ and $D_u^{(s)}$, respectively. A confidence mask is constructed based on the teacher prediction, and consistency is imposed only on high-confidence regions:

$$M_u(i,j) = \begin{cases} 1, & D_u^{(t)}(i,j) > \tau_u \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

where τ_u is adaptively determined from the sample-wise quantile of the teacher-predicted density map. In this work, the quantile parameter is set to q = 0.90, meaning that only the high-response regions of the teacher prediction are used for consistency learning. To avoid excessively sparse supervision, the minimum valid-mask ratio is further set to 0.02.

Based on this mask, the unlabeled density consistency loss is defined as a masked mean squared error:

$$\mathcal{L}_u^{den} = \frac{\sum_{i,j} M_u(i,j)\left(D_u^{(s)}(i,j) - D_u^{(t)}(i,j)\right)^2}{\sum_{i,j} M_u(i,j) + \epsilon_m} \tag{16}$$

where $\epsilon_m = 10^{-6}$ is a small constant introduced to avoid division by zero.

In addition to density map consistency, a count consistency constraint is also introduced. Let the teacher and student predicted counts be

$$\hat{C}_u^{(t)} = \sum_{i,j} D_u^{(t)}(i,j), \quad \hat{C}_u^{(s)} = \sum_{i,j} D_u^{(s)}(i,j) \tag{17}$$

Then, the count consistency loss for unlabeled samples is defined as

$$\mathcal{L}_u^{cnt} = \frac{1}{B}\sum_{b=1}^{B} \frac{\left|\hat{C}_{u,b}^{(s)} - \hat{C}_{u,b}^{(t)}\right|}{\hat{C}_{u,b}^{(t)} + \epsilon_c} \tag{18}$$

Thus, the final consistency loss for unlabeled samples is written as

$$\mathcal{L}_u = \mathcal{L}_u^{den} + \lambda_{ssl_{cnt}} \mathcal{L}_u^{cnt} \tag{19}$$

2.4.3. Total Loss

Combining labeled and unlabeled data, the total loss of this work is:

$$\mathcal{L}_{total} = \mathcal{L}_l + \alpha(e)\mathcal{L}_u \tag{20}$$

Among them, α(e) is the semi-supervised weight that varies with epoch. Adopting a linear warm-up strategy in implementation [20]:

$$\alpha(e) = \lambda_{ssl} \cdot \min\left(1, \frac{e+1}{T_{warm}}\right) \tag{21}$$

In the final training setting, $\lambda_{ssl} = 0.01$, $T_{warm} = 40$. These settings were further examined through sensitivity analysis in Section 3.5. This strategy allows the model to rely more on labeled supervision during the early stage of training and gradually increase the contribution of unlabeled consistency as training proceeds, so that pseudo-label information is introduced more stably after the teacher–student predictions become sufficiently reliable.

2.5. Baseline Methods and Implementation Details

To provide a fair and comprehensive comparison, representative baseline methods were selected from both semi-supervised counting and density regression counting models. For semi-supervised comparison, Dream [39], Calibrating [21], MTCP [40], and MRC [24] were used to evaluate counting performance under different labeled data ratios. These methods are closely related to semi-supervised density-map-based counting and are designed to reduce annotation dependency by exploiting unlabeled samples through pseudo-labeling, consistency regularization, uncertainty calibration, contextual modeling, or teacher–student learning strategies.

Dream is a semi-supervised crowd-counting method based on rank-consistent pyramid learning. It exploits the ranking relationship among density representations at different pyramid levels to improve the use of unlabeled data and enhance density map regression under limited annotations. Calibrating is an uncertainty-aware semi-supervised counting method that improves pseudo-label reliability by calibrating uncertainty estimation, thereby reducing the negative influence of noisy pseudo-supervision. MTCP, namely multi-task credible pseudo-label learning, introduces credible pseudo-label selection and multi-task learning into semi-supervised crowd counting. By improving the reliability of pseudo-labels and jointly optimizing related supervision signals, MTCP enhances counting robustness when only limited labeled samples are available. MRC is a semi-supervised crowd-counting method with contextual modeling. It aims to facilitate a more holistic understanding of dense scenes by modeling contextual information, thereby improving the model's ability to learn from unlabeled samples in complex counting scenarios.

For lightweight and deployment-oriented comparison, MCNN [15], TasselNetV2+ [41], and CSRNet [16] were selected. MCNN is a classical multi-column convolutional network for density-map-based counting. TasselNetV2+ is a compact regression-based counting model originally designed for plant counting tasks. CSRNet is a strong density regression baseline that uses dilated convolutions to enlarge the receptive field and has been widely used in dense object counting. These models were included to evaluate the accuracy–efficiency trade-off of M3DANet in terms of counting accuracy, parameter size, and inference speed.

All baseline models were evaluated on the same bee counting dataset using identical training, validation, and test splits. In the semi-supervised experiments, the validation and test sets were kept fixed across all labeled data ratios, while only the proportion of labeled samples in the training set was changed. The remaining training images were treated as unlabeled data. The same evaluation metrics, namely MAE and RMSE, were used for all methods.

3. Results

3.1. Experimental Setup

Most density-map-based counting studies adopt the mean absolute error (MAE) and root mean square error (RMSE) as standard evaluation metrics [42]. MAE measures the average absolute deviation between the predicted count and the ground truth, whereas RMSE penalizes large errors more strongly and is therefore more sensitive to outliers. In this work, MAE and RMSE are employed to evaluate the performance of the proposed semi-supervised bee colony counting model.

For a test set containing N images, let C_i and C_i^* denote the predicted and ground-truth numbers of bees in the i-th image, respectively. The MAE and RMSE are calculated as follows:

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|C_i - C_i^*| \tag{22}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(C_i - C_i^*\right)^2} \tag{23}$$

All the experiments were conducted in the same environment. The experimental environment and training configuration are as follows. The operating system is Ubuntu 20.04, with PyTorch 1.11.0, CUDA 11.3, and Python 3.8. The hardware platform includes an NVIDIA RTX 4090D GPU (24 GB; NVIDIA Corporation, Santa Clara, CA, USA), an Intel Xeon Platinum 8474C CPU (15 vCPUs; Intel Corporation, Santa Clara, CA, USA),and 80 GB RAM. The model is trained for 200 epochs with a batch size of 8 and a learning

rate of 1.0×10^{-5}. AdamW is used as the optimizer with weight decay of 1×10^{-4}. Semi-supervised training followed a teacher–student framework, in which the teacher parameters were updated from the student parameters using EMA with a decay factor of 0.999. The consistency loss included a density consistency term and a count-consistency term, weighted by $\lambda_{\text{ssl}} = 0.01$ and $\lambda_{ssl_{cnt}} = 0.01$, respectively. A 40-epoch warm-up strategy was adopted to gradually introduce the semi-supervised loss. These settings were used in the main experiments, and their influence was further evaluated in the sensitivity analysis.

All the comparative experiments followed a unified pipeline: input images were first resized proportionally and then randomly cropped to 768×768 during training; during validation and testing, sliding-window inference was performed with a 768×768 window and a stride of 384. Standard data augmentation, Gaussian density map supervision, AdamW weight decay, warm-up training, confidence masking, and fixed validation and test splits were used to improve training stability and reduce the risk of overfitting under limited labeled data.

3.2. Comparison of Counting Performance

As shown in Table 1, M3DANet achieved an MAE of 5.201 and an RMSE of 6.989 under the fully supervised setting. Compared with MCNN and TasselNetV2+, M3DANet substantially reduced both the MAE and RMSE, showing stronger robustness in dense bee counting scenes. Compared with the strong density regression baseline CSRNet, M3DANet obtained a slightly lower MAE (5.201 vs. 5.298), while CSRNet achieved a marginally lower RMSE (6.856 vs. 6.989). Therefore, the main advantage of M3DANet in the fully supervised setting is not a uniform improvement in every error metric, but a stronger accuracy–efficiency balance: it used only 2.095 M parameters, reduced the parameter size by approximately 87.1% relative to CSRNet, and achieved a 17.7-fold higher inference speed.

Table 1. A performance comparison of different methods on the main dataset under fully supervised and semi-supervised settings.

Model	Params (M)	FPS	Fully Supervised		10%		30%		50%	
			MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
MCNN	0.134	98.55	24.899	35.771	-	-	-	-	-	-
TasselNetV2+	0.260	258.38	9.593	12.787	-	-	-	-	-	-
CSRNet	16.26	23.57	5.298	6.856	-	-	-	-	-	-
Dream	16.26	278.79	-	-	11.597	14.350	9.915	14.329	8.809	12.034
Calibrating	21.95	1.11	-	-	14.616	19.995	8.565	12.223	8.122	10.166
MTCP	17.402	84.241	-	-	14.114	17.286	8.010	10.417	7.646	9.429
MRC	34.75	1.29	-	-	11.950	14.080	9.970	13.190	6.990	8.900
M3DANet (Ours)	2.095	416.64	5.201	6.989	9.937	13.093	7.003	9.387	5.570	7.620

Under the semi-supervised setting, M3DANet consistently obtained the lowest MAE and RMSE across the 10%, 30%, and 50% labeled data ratios. With only 10% labeled training images, M3DANet achieved an MAE of 9.937 and an RMSE of 13.093, outperforming Dream, Calibrating, MTCP, and MRC. At the 30% labeled data ratio, it further reduced the MAE to 7.003 and the RMSE to 9.387, remaining better than the closest baseline, MTCP. At the 50% labeled data ratio, M3DANet achieved the best semi-supervised result, with an MAE of 5.570 and an RMSE of 7.620. Compared with the strongest baseline at this ratio, MRC, the MAE and RMSE were reduced by 20.32% and 14.38%, respectively. These results indicate that the proposed teacher–student consistency strategy can effectively exploit unlabeled bee images and maintain stable gains under different annotation budgets.

From the annotation efficiency perspective, the MAE of M3DANet decreased from 9.937 to 7.003 and 5.570 as the labeled data ratio increased from 10% to 30% and 50%, and RMSE decreased from 13.093 to 9.387 and 7.620. The 50% semi-supervised setting was

close to the fully supervised result, with only 0.369 higher MAE and 0.631 higher RMSE. Notably, using only 50% labeled training images, M3DANet already achieved substantially lower counting errors than the fully supervised MCNN and TasselNetV2+, and approached the performance of the strong fully supervised density regression baseline CSRNet. This suggests that M3DANet can use unlabeled data to recover much of the performance that would otherwise require additional manual point annotations.

Figure 8 further shows that the predicted-count curve becomes progressively closer to the ground-truth curve as the labeled data ratio increases. Although the 10% setting already captures the overall count trend, larger deviations remain in high-count images. These deviations are reduced at 30% and further narrowed at 50%, confirming that the semi-supervised framework benefits from additional labeled samples while still preserving strong performance under limited annotation.

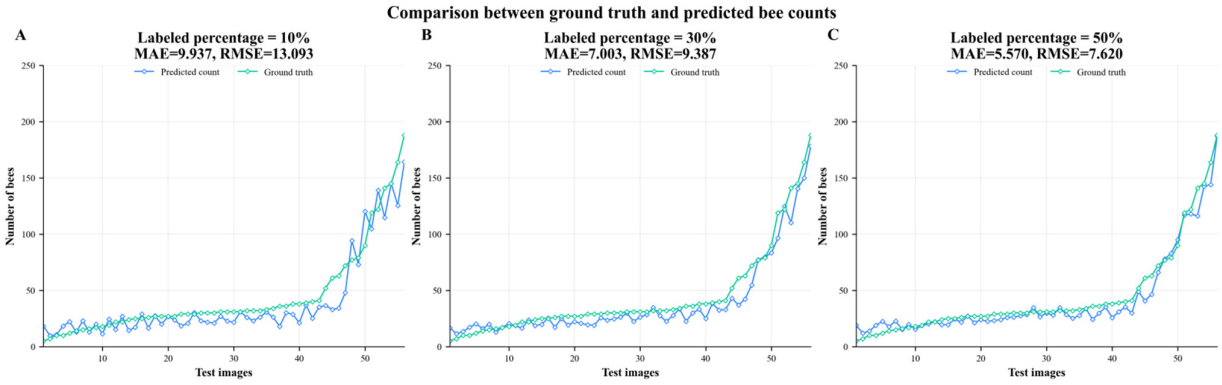

Figure 8. Comparison between ground-truth and predicted bee counts for M3DANet at labeled percentages of 10%, 30%, and 50%.

As shown in Figure 9, the further qualitative comparison indicates that, with the ground-truth count (GT) for this sample being 135, M3DANet produces more complete responses in bee-dense regions, reduces missed responses, and provides a predicted count closer to the ground truth than the other semi-supervised methods.

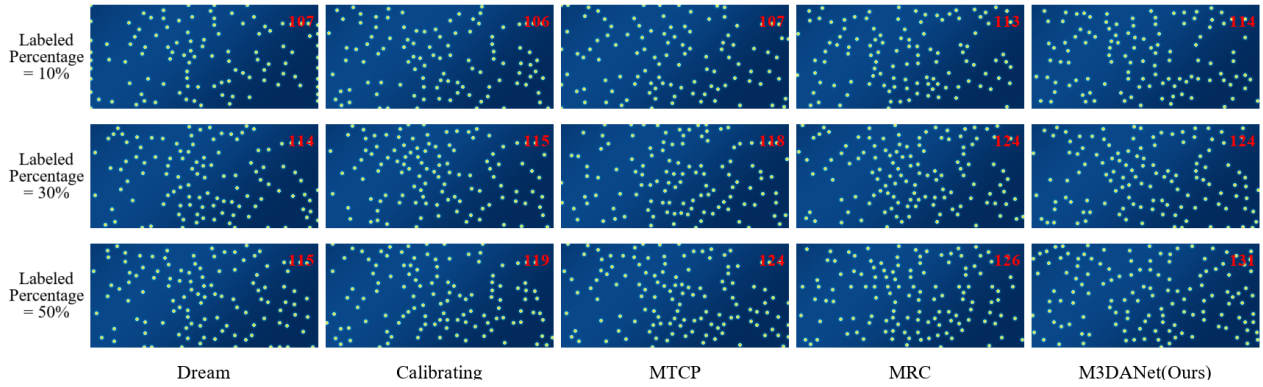

Figure 9. Qualitative comparison of predicted responses and counts among semi-supervised methods. Rows correspond to labeled data ratios of 10%, 30%, and 50%; columns correspond to Dream, Calibrating, MTCP, MRC, and M3DANet.

3.3. Ablation Experiments

3.3.1. Structural Ablation Experiment

The structural ablation results are shown in Table 2. BackboneOnly, consisting of the MobileNetV3-Large backbone and a lightweight density regression head, was used as the

baseline. On this basis, the multi-scale context encoding module (MSCE), the attention-guided low-level fusion module (AGLF), and their combination were introduced to evaluate their individual and joint effects on counting performance.

Table 2. The results of the structural ablation experiment.

BackboneOnly	MSCE	AGLF	MAE	RMSE	*p*-Value vs. Full
✓	×	×	7.064	9.194	0.0340
✓	✓	×	6.455	8.761	0.0156
✓	×	✓	5.922	7.501	0.0216
✓	✓	✓	5.201	6.989	-

BackboneOnly achieved an MAE of 7.064 and an RMSE of 9.194. After introducing MSCE, the MAE and RMSE decreased to 6.455 and 8.761, corresponding to relative reductions of 8.62% and 4.71%, respectively. This indicates that multi-scale contextual encoding improves the representation of bee targets with different apparent sizes and enhances density regression in complex scenes.

When only AGLF was added, the MAE and RMSE further decreased to 5.922 and 7.501, with relative reductions of 16.17% and 18.41% compared with BackboneOnly. This result suggests that coordinate attention and low-level feature fusion are effective for preserving boundary cues, local textures, and spatial location information, thereby improving detail recovery in dense regions and reducing background interference.

When MSCE and AGLF were jointly enabled, the full M3DANet achieved the best performance, with an MAE of 5.201 and an RMSE of 6.989. Compared with BackboneOnly, the full model reduced MAE and RMSE by 26.37% and 23.98%, respectively. The paired Wilcoxon signed-rank tests showed that the full model significantly outperformed all three reduced structural variants ($p < 0.05$), confirming that MSCE and AGLF are complementary in multi-scale contextual modeling and low-level spatial detail restoration.

3.3.2. Semi-Supervised Component Ablation Experiment

To further verify the effectiveness of the semi-supervised learning strategy, an additional component ablation experiment was conducted under the 50% labeled data setting. The complete framework contains density consistency, count consistency, confidence masking, and the warm-up strategy. The results are reported in Table 3.

Table 3. The results of the semi-supervised component ablation experiment.

Variant	Density Consistency	Count Consistency	Mask	Warm-Up	MAE	RMSE	*p*-Value
Full SSL	✓	✓	✓	✓	5.570	7.620	-
No SSL	×	×	-	-	6.603	8.912	0.0036
Count-only	×	✓	✓	✓	6.801	9.223	0.0287
Density-only	✓	×	✓	✓	7.192	9.710	0.0026
No mask	✓	✓	×	✓	7.004	9.426	0.0100
No warm-up	✓	✓	✓	×	6.994	9.413	0.0017

The full semi-supervised framework achieved the best result, with an MAE of 5.570 and an RMSE of 7.620. When the consistency constraints were removed, the model degenerated into a supervised-only setting using labeled samples only, and the MAE and RMSE increased to 6.603 and 8.912, respectively. Compared with this setting, the full semi-supervised framework reduced the MAE and RMSE by 15.64% and 14.50%, respectively, indicating that unlabeled samples provide useful supplementary supervision through teacher–student consistency learning.

The single-consistency variants were also inferior to the full framework. The Count-only variant obtained an MAE of 6.801 and an RMSE of 9.223, whereas the Density-only variant obtained an MAE of 7.192 and an RMSE of 9.710. These results demonstrate the complementarity between pixel-level density consistency and image-level count consistency. Density consistency constrains the spatial distribution of the predicted density map, while count consistency constrains the global counting result.

Removing the confidence mask increased the MAE and RMSE to 7.004 and 9.426, respectively, showing that directly using all pseudo-density responses may introduce noise from low-confidence regions. Removing the warm-up strategy also increased the MAE and RMSE to 6.994 and 9.413, respectively, suggesting that gradually introducing the semi-supervised loss during early training prevents unstable teacher predictions from imposing excessive incorrect supervision. All reduced semi-supervised variants differed significantly from the full SSL framework in paired Wilcoxon tests ($p < 0.05$).

Figure 10 summarizes the ablation results. The structural ablation results show that the model error decreases progressively with the introduction of MSCE and AGLF, while the semi-supervised component ablation shows that removing any key component increases the error. These findings verify the effectiveness of both the proposed network structure and the semi-supervised training strategy.

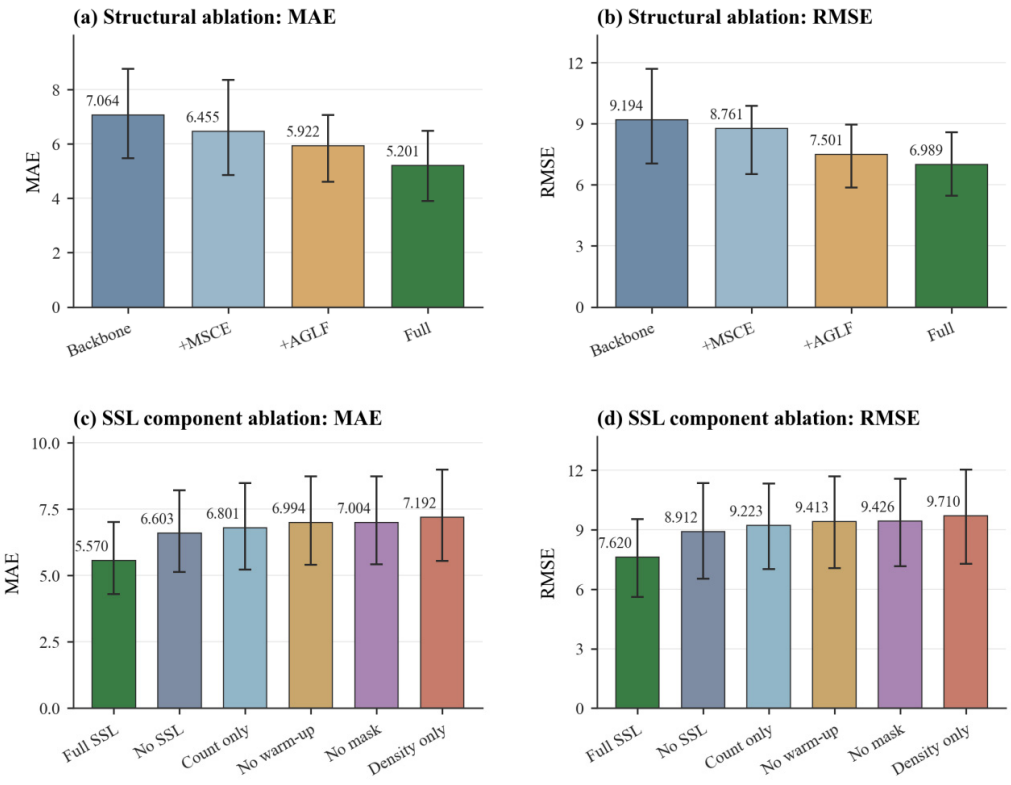

Figure 10. A quantitative visualization of the structural and semi-supervised component ablation experiments. The error bars indicate bootstrap 95% confidence intervals.

3.4. Generalization Experiment

An independent self-built dataset was used to evaluate the generalization ability of the model across different honeybee species. The main dataset consists mainly of images of Italian bees, while the generalized dataset comes from Chinese bees, and the differences between the two species may lead to domain shift. Therefore, this dataset provides a suitable test case for the robustness of cross-species counting. The annotation and preprocessing process of the generalized dataset is consistent with that of the main dataset.

This dataset contains a total of 504 images and 133,888 annotation points. The experimental configuration for generalization testing is the same as that used in the main experiment; therefore, network inputs, training strategies, inference processes, and evaluation metrics will not be repeated here. Figure 11 shows a visualization example of a general dataset. The preprocessed image, point annotation visualization, and corresponding generated density map are displayed in sequence.

Figure 11. Examples from the Chinese honeybee generalization dataset. (**A**) Original image; (**B**) Point annotation visualization, where the red dots indicate manually annotated bee locations; (**C**) Generated density map, where warmer colors indicate higher local bee-density responses.

As shown in Table 4, Bias denotes the mean signed prediction error and is used to identify whether a model tends to systematically overestimate or underestimate image-level counts. Under the 10% labeled setting, M3DANet achieved the lowest MAE and RMSE, which were 13.947 and 17.649, respectively. Compared with Dream, Calibrating, MTCP, and MRC, the MAE of M3DANet was reduced by 49.7%, 60.3%, 44.8%, and 14.7%, respectively. This indicates that M3DANet can use limited supervision more effectively and maintain stronger cross-species generalization on Chinese honeybee images.

Under the 30% labeled setting, M3DANet achieved an MAE of 11.945 and an RMSE of 14.078, showing the best performance among all semi-supervised methods at this labeled ratio. Compared with Dream, Calibrating, MTCP, and MRC, the MAE of M3DANet was reduced by 34.8%, 44.4%, 36.2%, and 8.8%, respectively. Under the 50% labeled setting, the MAE and RMSE of M3DANet were 11.772 and 15.893, respectively, remaining lower than those of Dream, Calibrating, and MTCP. MRC achieved an MAE of 11.300 and an RMSE of 14.237, which were numerically lower than those of M3DANet.

Under the fully supervised setting, the MAE and RMSE of M3DANet were 11.317 and 14.390, respectively, outperforming MCNN and TasselNetV2+ and approaching CSR-Net. The Bias values show that Dream, Calibrating, and MTCP had clear negative bias on the Chinese honeybee dataset, especially under the 10% labeled setting, where their Bias values were −22.756, −24.231, and −24.998, respectively. In contrast, the Bias values of M3DANet under the 10% and 50% labeled settings were −5.086 and −0.859, respectively, indicating that M3DANet alleviated systematic underestimation in the cross-species generalization scenario.

Figure 12 further shows that M3DANet has a lower and more compact per-image error distribution than Dream, Calibrating, and MTCP, especially under the 10% and 30% labeled settings.

Table 4. A performance comparison of different methods on the Chinese honeybee generalization dataset.

Labeled Percentage	Model	MAE	RMSE	Bias
Full	MCNN	16.638	21.753	−12.499
	TasselNetV2+	15.539	19.336	−13.666
	CSRNet	11.538	15.055	−2.651
	M3DANet (Ours)	11.317	14.390	−2.776
10%	Dream	27.746	33.141	−22.756
	Calibrating	35.154	45.070	−24.231
	MTCP	25.277	29.328	−24.998
	MRC	16.358	19.242	−13.320
	M3DANet (Ours)	13.947	17.649	−5.086
30%	Dream	18.314	21.634	−16.544
	Calibrating	21.491	25.693	−19.550
	MTCP	18.724	23.180	−13.193
	MRC	13.104	16.589	−6.435
	M3DANet (Ours)	11.945	14.078	−5.414
50%	Dream	13.075	16.332	−5.912
	Calibrating	13.947	17.649	−5.086
	MTCP	15.565	20.888	−13.752
	MRC	11.300	14.237	−1.915
	M3DANet (Ours)	11.772	15.893	−0.859

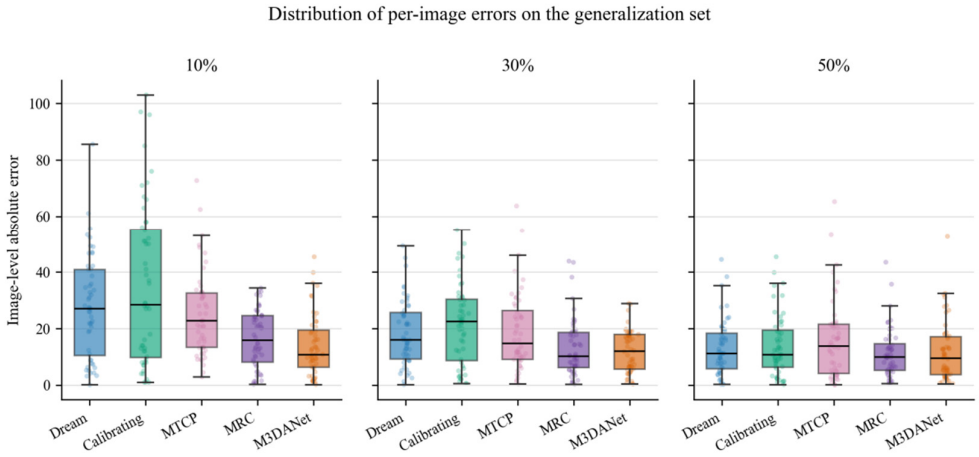

Figure 12. The distribution of per-image absolute errors on the Chinese honeybee generalization dataset.

To further determine whether the observed differences were statistically reliable, paired significance analysis was performed based on image-level predictions. For each test image, the absolute error of M3DANet was compared with that of each baseline model, and the paired improvement was calculated as the baseline absolute error minus the M3DANet absolute error. Therefore, a positive paired improvement indicates that M3DANet produced a lower absolute error than the corresponding baseline on the same image. Because image-level errors may not follow a normal distribution, the Wilcoxon signed-rank test was used for paired testing, and the Holm–Bonferroni method was applied for multiple-comparison correction.

Figure 13 indicates that M3DANet significantly outperforms Dream, Calibrating, and MTCP under the 10% and 30% labeled settings after Holm correction. The paired difference between M3DANet and MRC is not significant at the 30% and 50% labeled

settings, although M3DANet has a lower mean error at 30% and MRC is slightly lower at 50%.

Figure 13. Paired error improvement of M3DANet over each semi-supervised baseline. Positive values indicate lower absolute error for M3DANet.

Overall, the Chinese honeybee generalization experiment demonstrates that M3DANet has strong low-label cross-species generalization ability. In particular, M3DANet achieved the lowest MAE and RMSE among all semi-supervised methods under the 10% and 30% labeled settings. The paired tests further show significant improvements over Dream, Calibrating, and MTCP at these two labeled ratios. After including MRC, the differences between M3DANet and MRC under the 30% and 50% labeled settings were not statistically significant, indicating comparable performance when more labeled data are available. These results suggest that M3DANet is not only applicable to Italian honeybee images in the main dataset but can also transfer to Chinese honeybee images while maintaining stable counting performance under low-label cross-species conditions.

3.5. Hyperparameter Sensitivity and Stability Analysis

To further address the concern regarding the stability and justification of key hyperparameter settings, a sensitivity analysis was conducted for three parameters that directly affect the semi-supervised learning process: the semi-supervised consistency-loss weight λ_{ssl}, the exponential moving average (EMA) decay coefficient β, and the confidence-mask quantile q. All the experiments were performed under the 50% labeled data setting. A one-factor-at-a-time strategy was adopted: unless otherwise specified, the reference configuration was fixed as $\lambda_{ssl} = 0.01$, $\beta = 0.999$, and $q = 0.90$, and only the target hyperparameter was varied in each group. The remaining network architecture, data split, preprocessing strategy, and training protocol were kept unchanged. The results are summarized in Table 5.

As shown in Table 5, the repeated reference configuration obtained identical results in the three hyperparameter groups, with an MAE of 5.570 and an RMSE of 7.620, confirming that the experiments followed a controlled single-variable design. For the semi-supervised consistency-loss weight λ_{ssl}, the best performance was achieved at $\lambda_{ssl} = 0.01$. A smaller value weakened the use of unlabeled samples, whereas a larger value may introduce pseudo-label noise or over-constrain the student network.

For the EMA decay coefficient ρ, the lowest MAE and RMSE were obtained when $\rho = 0.999$. A smaller value caused the teacher model to update too rapidly and reduced

pseudo-supervision stability, while a larger value made the teacher model adapt too slowly to the student network. Therefore, ρ = 0.999 provided a suitable balance between temporal smoothing and update responsiveness.

Table 5. Sensitivity analysis of key hyperparameters.

Analyzed Hyperparameter	Value	MAE	RMSE
Semi-supervised consistency-loss weight λ_{ssl}	0.005	7.003	9.387
	0.01	5.570	7.620
	0.02	6.422	9.356
EMA decay coefficient ρ	0.990	6.994	9.247
	0.999	5.570	7.620
	0.9995	7.003	9.387
Confidence-mask quantile q	0.80	6.400	8.222
	0.90	5.570	7.620
	0.95	8.777	12.008

For the confidence-mask quantile q, the best result was obtained at q = 0.90. A lower threshold may include low-confidence background responses, whereas a higher threshold may discard useful high-response regions. Overall, the sensitivity analysis supports the final settings of λ_{ssl} = 0.01, ρ = 0.999, and q = 0.90, demonstrating the empirical rationality and robustness of the proposed semi-supervised training strategy.

3.6. Deployment Verification

M3DANet effectively reduced model complexity while maintaining high counting accuracy, making it suitable for deployment in resource-constrained real-world bee monitoring scenarios. To further evaluate its deployment feasibility, the trained model was deployed on a self-developed handheld bee-counting device based on the NVIDIA Jetson Orin NX platform (NVIDIA Corporation, Santa Clara, CA, USA). As shown in Figure 14, the device was tested in a real beekeeping environment under natural lighting conditions. Figure 14A and 14B show the collection process for the edge-area samples and the whole-frame samples, respectively; Figure 14C shows close-range acquisition of a bee cluster region, and Figure 14D shows the on-site collection process under handheld operation. The main deployment configuration and edge device benchmark results are shown in Table 6. The deployed M3DANet model contains only 2.095 million parameters. On the Jetson Orin NX platform, the average inference latency was 65.75 ms/image, and the throughput of the complete processing flow was 10.44 FPS. The average process memory usage was 943.51 MB, further demonstrating the lightweight nature of M3DANet and its ability to run on portable edge hardware with moderate computational and memory overhead.

Table 6. Edge deployment benchmark of M3DANet on Jetson Orin NX.

Platform	Parameters (M)	Model Size (MB)	Input Size	Latency (ms)	FPS	Memory (MB)	Power (W)
Jetson Orin NX	2.095	8.07	Long side = 1280 px	65.75	10.44	943.51	10.00

Figure 15 presents two representative visualization examples of field counting results under real-world variations. The first row shows a close-range bee cluster sample, and the second row shows a whole-frame bee colony sample with a more complex hive background. In both cases, the high-response regions of the predicted density maps are generally consistent with the actual bee cluster distributions, indicating that the model can effectively focus on bee-populated areas while suppressing background interference. The

predicted counts for the two cases are 262 and 211, respectively, which further supports the real-time counting capability and deployment feasibility of M3DANet in practical beekeeping scenarios.

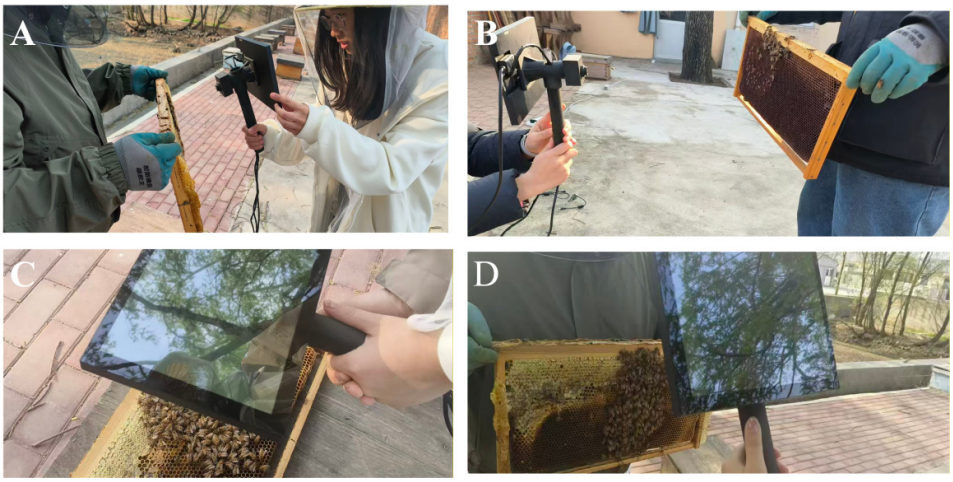

Figure 14. Edge deployment and real-time visualization of M3DANet. (**A**) Edge-area sample acquisition in a real beekeeping environment. (**B**) Whole-frame sample acquisition. (**C**) Close-range acquisition of a bee cluster region with real-time visualization. (**D**) On-site handheld operation under natural lighting conditions.

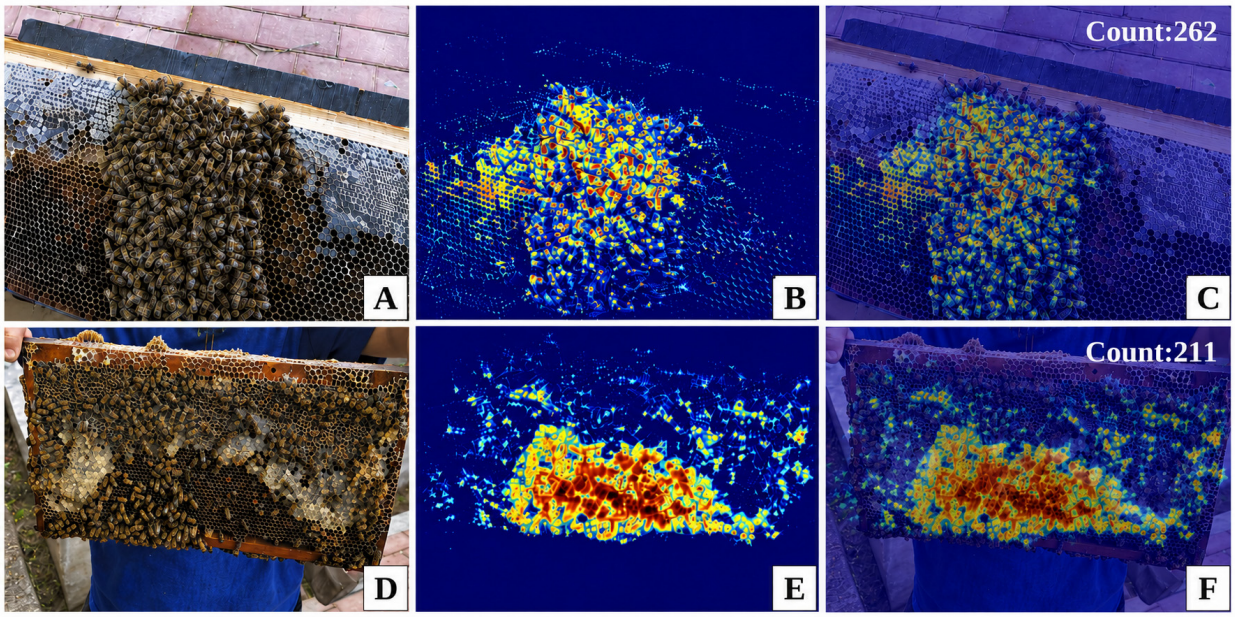

Figure 15. Field visualization examples of bee counting results under real-world variations. (**A**,**D**) Original field images; (**B**,**E**) predicted density heatmaps; (**C**,**F**) overlays of the predicted density heatmaps and original images. The colors in the heatmaps indicate the predicted bee-density responses, with warmer colors representing higher-density regions. The first row shows a close-range bee cluster sample, and the second row shows a whole-frame bee colony sample with a more complex background.

3.7. Failure Case Analysis

To further clarify the robustness boundary of the proposed method, a failure case analysis was added on the fixed test, where the SSL loss weight was 0.01 and 50% of the training images were labeled. Since M3DANet is a density regression model rather than an instance detector, image-level under-counting and over-counting were used to approximate missed-detection and false-positive-like tendencies, respectively. On the 58 test images,

the model achieved an MAE of 5.570 and an RMSE of 7.620, with under-counting cases being more frequent than over-counting cases, indicating that missed detections were the dominant error pattern. As shown in Figure 16, the under-counting case mainly occurs in a high-density bee cluster, where adjacent and partially occluded bees generate merged or weakened density responses, resulting in a lower predicted count than the ground truth. In contrast, over-counting tends to appear in sparse scenes, where honeycomb texture and visually similar local background patterns may introduce weak false-positive-like density responses. These observations indicate that M3DANet performs reliably on the fixed test set of the main dataset, but its remaining errors are mainly associated with dense occlusion and appearance ambiguity. Future work will therefore focus on robustness-oriented augmentation and uncertainty-aware pseudo-label filtering to further reduce missed counts in highly crowded field scenes.

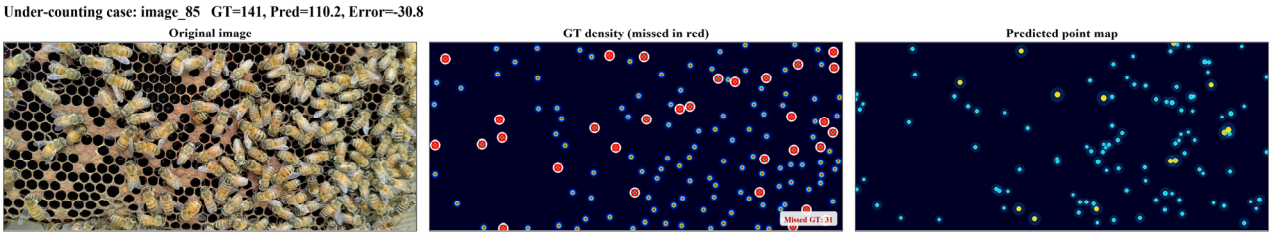

Figure 16. A representative under-counting failure case on the main test set. The middle panel shows the manually annotated ground-truth bee points. Red circled crosses mark missed ground-truth bee points, namely annotated bee locations that have no corresponding predicted point in the prediction map. The right panel shows the predicted bee points; cyan and yellow points indicate predicted responses, with yellow denoting stronger responses.

4. Discussion

4.1. Overall Counting Performance: Synergy of Architecture and Semi-Supervised Strategy

Under the fully supervised setting (Table 1), M3DANet achieved an MAE of 5.201 and an RMSE of 6.989. Compared with the lightweight models MCNN (24.899 and 35.771) and TasselNetV2+ (9.593 and 12.787), M3DANet substantially reduced the counting errors, indicating that the multi-scale context encoding (MSCE) and attention-guided low-level fusion (AGLF) modules can effectively handle scale variation and background clutter in bee images. Compared with the strong baseline CSRNet (5.298 and 6.856), M3DANet achieved a slightly lower MAE (−1.8%) and a slightly higher RMSE (+1.9%), while using only 12.9% of the parameters (2.095 M vs. 16.26 M) and achieving a 17.7 times higher inference speed (416.6 FPS vs. 23.6 FPS). Therefore, the main advantage of M3DANet is not uniform superiority over heavier models, but a stronger accuracy–efficiency trade-off.

Under the semi-supervised settings with 10%, 30%, and 50% labeled data ratios, M3DANet consistently achieved the lowest MAE and RMSE at all ratios (Table 1). For example, at the 10% labeled data ratio, the MAE of M3DANet was 9.937, which was 14.3% lower than DREAM. At the 50% labeled data ratio, the MAE was 5.570, which was 20.3% lower than MRC. This consistent advantage comes from the joint effect of the network architecture and the semi-supervised learning strategy. On the one hand, MSCE and AGLF provide feature representations suitable for dense bee counting scenes. On the other hand, the complementary density and count consistency losses, together with confidence masking and warm-up training, enable the model to effectively exploit unlabeled samples.

4.2. Analytical Interpretation of Module Contributions: Complementarity of MSCE and AGLF

The structural ablation experiment (Table 2) shows that the backbone-only model achieved an MAE of 7.064. Adding MSCE reduced the MAE to 6.455, corresponding to a

relative reduction of 8.6%. Adding AGLF further reduced the MAE to 5.922, corresponding to a relative reduction of 16.2%. When both modules were combined, the MAE was further reduced to 5.201, corresponding to a relative reduction of 26.4%. Paired Wilcoxon tests confirmed that the full model significantly outperformed all reduced variants ($p < 0.05$).

The improvement from MSCE can be attributed to its ability to capture sparse individuals, local clusters, and highly crowded regions simultaneously through atrous spatial pyramid pooling and multi-scale dilated convolutions. This ability is important because bee images usually show highly non-uniform density distributions. The improvement from AGLF is more pronounced because high-level semantic features tend to lose boundary and texture details after downsampling, while AGLF recovers spatial location cues of adjacent or partially occluded bees through coordinate attention and low-level feature fusion. It also suppresses false-positive background responses caused by honeycomb texture, wooden frame edges, and shadows. Therefore, MSCE and AGLF are complementary, and they jointly achieve multi-scale context modeling and low-level spatial detail restoration.

4.3. Mechanism of the Semi-Supervised Strategy: Complementarity of Consistency Losses and Stabilization

The semi-supervised ablation experiment (Table 3, 50% labeled data ratio) further reveals the role of each component. Removing the semi-supervised loss entirely increased the MAE to 6.603, while the full framework reduced the MAE to 5.570, corresponding to a relative reduction of 15.6%. This result demonstrates that unlabeled samples can provide effective supplementary supervision through teacher–student consistency learning.

Using only count consistency or only density consistency resulted in MAE values of 6.801 and 7.192, respectively, and both were significantly higher than the joint version (5.570). This directly verifies their complementarity. Density consistency constrains the spatial distribution of the predicted density map, while count consistency constrains the global count. Together, they prevent the model from producing density maps that are visually plausible but numerically biased. Removing the confidence mask increased the MAE to 7.004, indicating that low-confidence pseudo-responses can introduce noise. Removing the warm-up strategy increased the MAE to 6.994, suggesting that imposing consistency loss too early may amplify errors when the teacher model is still unreliable. All the variants differed significantly from the full framework ($p < 0.05$), confirming the necessity of these designs for stabilizing teacher–student learning.

4.4. Generalization Ability and Labeling Efficiency

The cross-species generalization experiment on the Chinese honeybee dataset further evaluates the transferability of M3DANet under species and background shifts. Under the fully supervised setting, M3DANet achieved an MAE of 11.317 and an RMSE of 14.390, outperforming MCNN and TasselNetV2+ and approaching CSRNet, while maintaining a much smaller model size and higher inference efficiency. Under the semi-supervised settings, the advantage of M3DANet was more evident when labeled data were limited. With only 10% labeled training images, M3DANet achieved an MAE of 13.947 and an RMSE of 17.649, outperforming Dream, Calibrating, MTCP, and MRC. At the 30% labeled data ratio, it achieved the best semi-supervised performance, with an MAE of 11.945 and an RMSE of 14.078. These results indicate that the combination of multi-scale feature representation, attention-guided low-level fusion, and density and count consistency learning helps M3DANet exploit unlabeled images more effectively under cross-species domain shifts and reduces systematic underestimation.

From the labeling-efficiency perspective, the main-dataset results show that the MAE of M3DANet decreased from 9.937 to 7.003 and 5.570 as the labeled data ratio increased from 10% to 30% and 50%, while the RMSE decreased from 13.093 to 9.387 and 7.620.

The improvement from 10% to 30% was larger than that from 30% to 50%, suggesting a diminishing-return trend as more labeled images were added. Notably, the 50% semi-supervised setting approached the fully supervised result, with only 0.369 higher MAE and 0.631 higher RMSE. Moreover, using only 50% labeled training images, M3DANet already achieved substantially lower counting errors than the fully supervised MCNN and TasselNetV2+ and approached the performance of CSRNet. These findings demonstrate that M3DANet can recover much of the performance that would otherwise require additional manual point annotations, providing a practical trade-off among annotation cost, counting accuracy, generalization ability, and deployment feasibility.

4.5. Limitations

Despite the encouraging results, several limitations remain.

(1) Sensitivity to extreme density conditions: As shown in the failure case analysis (Section 3.7), adjacent bees in highly crowded and severely occluded regions may produce merged or weakened density responses, which can lead to under-counting. When the local density exceeds the maximum density range observed in the training set, the prediction bias may increase systematically.

(2) Dataset constraints: Although the main dataset covers representative variations in bee density, shooting distance, illumination, viewing angle, and hive background conditions, its scale and environmental diversity remain limited compared with long-term real-world beekeeping conditions. In the Chinese honeybee generalization experiment, MRC performed similarly to M3DANet at the 50% labeled data ratio, indicating that cross-species generalization is still constrained by the coverage of training data.

(3) Dependence on pseudo-label quality: The ablation experiments show that removing the confidence mask or the warm-up strategy significantly increased the MAE to approximately 7.0, indicating that pseudo-label noise can still affect the student model. When the labeled data ratio is extremely low or when the unlabeled data distribution differs greatly from the labeled data distribution, such noise becomes harder to suppress.

Future work will focus on constructing larger and more diverse datasets, incorporating uncertainty-aware pseudo-label filtering, and designing adaptive inference strategies for extreme-density scenes.

5. Conclusions

In dense bee counting scenarios, manual point-level annotation is labor-intensive and deployment resources are often limited, making it challenging to balance counting accuracy, annotation efficiency, and practical deployability. To address this problem, this study proposes M3DANet, a lightweight semi-supervised bee counting method. The model is built on a MobileNetV3-Large backbone and incorporates a multi-scale context encoding (MSCE) module to capture density variations from sparse individuals to highly crowded clusters, as well as an attention-guided low-level fusion (AGLF) module to recover fine spatial details and suppress background interference. For semi-supervised learning, a teacher–student framework with density consistency, count consistency, confidence masking, and a warm-up strategy is employed to effectively exploit unlabeled images under limited annotation budgets.

Extensive experiments demonstrate that M3DANet consistently outperforms representative semi-supervised methods, including DREAM, Calibrating, MTCP, and MRC, at labeled ratios of 10%, 30%, and 50%. For example, at 10% labeling, M3DANet achieves an MAE of 9.937, which is 14.3% lower than DREAM, and at 50% labeling, the MAE is 5.570,

which is 20.3% lower than MRC. Under the fully supervised setting, M3DANet attains an MAE of 5.201 and an RMSE of 6.989 with only 2.095 M parameters and an inference speed of 416.64 FPS, achieving a strong accuracy–efficiency trade-off compared with heavier models such as CSRNet and more compact ones such as MCNN and TasselNetV2+. Edge deployment on a Jetson Orin NX device further confirms its real-time counting capability and engineering feasibility in real beekeeping environments.

Overall, M3DANet provides an effective solution for automated bee counting that jointly considers counting accuracy, annotation efficiency, and deployability. Future work will focus on expanding data diversity across seasons, hive types, and bee species, incorporating temporal modeling from video sequences, improving semi-supervised pseudo-label filtering with uncertainty estimation, and conducting long-term stability evaluation under continuous outdoor operation.

Author Contributions: Conceptualization, Z.L. and G.W.; methodology, X.L.; software, X.L. and M.M.; validation, X.L. and Y.K.; formal analysis, X.L.; investigation, H.H., Q.L. and F.L.; resources, H.H., Q.L. and F.L.; data curation, X.L., M.M. and Y.K.; writing—original draft preparation, X.L., M.M. and Y.K.; writing—review and editing, F.L., Z.L. and G.W.; visualization, X.L. and Y.K.; supervision, Z.L. and G.W.; project administration, Z.L. and G.W.; funding acquisition, Z.L. and F.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation Project of Shandong Province, grant number ZR2024MC173; the earmarked fund for Jiangxi Agriculture Research System, grant number JXARS-13.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Klein, A.-M.; Vaissière, B.E.; Cane, J.H.; Steffan-Dewenter, I.; Cunningham, S.A.; Kremen, C.; Tscharntke, T. Importance of pollinators in changing landscapes for world crops. *Proc. R. Soc. B-Biol. Sci.* **2006**, *274*, 303–313. [CrossRef] [PubMed]
2. Potts, S.G.; Biesmeijer, J.C.; Kremen, C.; Neumann, P.; Schweiger, O.; Kunin, W.E. Global pollinator declines: Trends, impacts and drivers. *Trends Ecol. Evol.* **2010**, *25*, 345–353. [CrossRef]
3. Nguyen, D.T.; Le, T.N.; Phung, T.H.; Nguyen, D.M.; Nguyen, H.Q.; Pham, H.T.; Phan, T.T.H.; Vu, H.; Le, T.L. Improving pollen-bearing honey bee detection from videos captured at hive entrance by combining deep learning and handling imbalance techniques. *Ecol. Inform.* **2024**, *82*, 102744. [CrossRef]
4. Borlinghaus, P.; Odemer, R.; Tausch, F.; Schmidt, K.; Grothe, O. Honey bee counter evaluation-Introducing a novel protocol for measuring daily loss accuracy. *Comput. Electron. Agric.* **2022**, *197*, 106957. [CrossRef]
5. Ratnayake, A.M.B.; Suhaimi, H.; Abas, P.E. Transforming Beekeeping Through Technology: A Systematic Review of Precision Beekeeping. *Sci* **2026**, *8*, 87. [CrossRef]
6. Sledevič, T.; Matuzevičius, D. Labeled dataset for bee detection and direction estimation on entrance to beehive. *Data Brief.* **2024**, *52*, 110060. [CrossRef]
7. Padubidri, C.; Kamilaris, A.; Charalambous, A.; Lanitis, A.; Constantinides, M. The Be-Hive Project-Counting Bee Traffic Based on Deep Learning and Pose Estimation. In *Intelligent Systems and Applications: Proceedings of the 2023 Intelligent Systems Conference (IntelliSys 2023)*; Lecture Notes in Networks and Systems; Springer Nature: Cham, Switzerland, 2024; pp. 531–545. [CrossRef]
8. Kongsilp, M.; Taetragool, U.; Duangphakdee, O. Individual honey bee tracking in beehive using deep learning and Kalman filter. *Sci. Rep.* **2024**, *14*, 1061. [CrossRef]
9. Bozek, K.; Hebert, L.; Portugal, Y.; Mikheyev, A.S.; Stephens, G.J. Markerless tracking of an entire honey bee colony. *Nat. Commun.* **2021**, *12*, 1733. [CrossRef]
10. Bilik, S.; Zemcik, T.; Kratochvila, L.; Ricanek, D.; Richter, M.; Zambanini, S.; Horak, K. Machine learning and computer vision techniques in continuous beehive monitoring applications: A survey. *Comput. Electron. Agric.* **2024**, *217*, 108560. [CrossRef]
11. Rozenbaum, E.; Shrot, T.; Daltrophe, H.; Kunya, Y.; Shafir, S. Machine learning-based bee recognition and tracking for advancing insect behavior research. *Artif. Intell. Rev.* **2024**, *57*, 245. [CrossRef]

12. Bjerge, K.; Mann, H.M.R.; Hoye, T.T. Real-time insect tracking and monitoring with computer vision and deep learning. *Remote Sens. Ecol. Conserv.* **2022**, *8*, 315–327. [CrossRef]
13. Bereciartua-Pérez, A.; Gómez, L.; Picón, A.; Navarra-Mestre, R.; Klukas, C.; Eggers, T. Insect counting through deep learning-based density maps estimation. *Comput. Electron. Agric.* **2022**, *197*, 106933. [CrossRef]
14. Lempitsky, V.; Zisserman, A. Learning to count objects in images. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems (NeurIPS 2010), Vancouver, BC, Canada, 6–11 December 2010*; Curran Associates, Inc.: Red Hook, NY, USA, 2010; pp. 1309–1317. [CrossRef]
15. Zhang, Y.; Zhou, D.; Chen, S.; Gao, S.; Ma, Y. Single-image crowd counting via multi-column convolutional neural network. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016*; IEEE: Piscataway, NJ, USA, 2016; pp. 589–597. [CrossRef]
16. Li, Y.; Zhang, X.; Chen, D. CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 1091–1100. [CrossRef]
17. Liu, Y.B.; Cao, G.; Shi, H.; Hu, Y.X. Lw-Count: An effective lightweight encoding-decoding crowd counting network. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 6821–6834. [CrossRef]
18. Odemer, R. Approaches, challenges, and recent advances in automated bee counting devices—A review. *Ann. Appl. Biol.* **2022**, *180*, 73–89. [CrossRef]
19. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (NeurIPS 2017)*; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 1195–1204. [CrossRef]
20. Berthelot, D.; Carlini, N.; Cubuk, E.D.; Kurakin, A.; Sohn, K.; Zhang, H.; Raffel, C. ReMixMatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv* **2020**, arXiv:1911.09785. [CrossRef]
21. Li, C.; Hu, X.; Abousamra, S.; Chen, C. Calibrating uncertainty for semi-supervised crowd counting. In *Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 1–6 October 2023*; IEEE: Piscataway, NJ, USA, 2023; pp. 1524–1533. [CrossRef]
22. Chen, J.; Wang, Z. Multi-task semi-supervised crowd counting via global to local self-correction. *Pattern Recognit.* **2023**, *140*, 109506. [CrossRef]
23. Wei, X.; Qiu, Y.; Ma, Z.; Hong, X.; Gong, Y. Semi-Supervised Crowd Counting via Multiple Representation Learning. *IEEE Trans. Image Process.* **2023**, *32*, 5220–5230. [CrossRef]
24. Qian, Y.; Hong, X.; Guo, Z.; Arandjelović, O.; Donovan, C.R. Semi-Supervised Crowd Counting With Contextual Modeling: Facilitating Holistic Understanding of Crowd Scenes. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 8230–8241. [CrossRef]
25. Alaba, S.; Shah, C.; Nabi, M.M.; Ball, J.; Moorhead, R.; Han, D.; Prior, J.; Campbell, M.; Wallace, F. Semi-supervised learning for fish species recognition. In *Proceedings of the Ocean Sensing and Monitoring XV, Orlando, FL, USA, 3–4 May 2023*; SPIE: Bellingham, WA, USA, 2023; Volume 12543, p. 1254326. [CrossRef]
26. Sohn, K.; Berthelot, D.; Carlini, N.; Zhang, Z.; Zhang, H.; Raffel, C.; Cubuk, E.D.; Kurakin, A.; Li, C.-L. FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence. In *Proceedings of the Advances in Neural Information Processing Systems 33 (NeurIPS 2020), Virtual, 6–12 December 2020*; Curran Associates, Inc.: Red Hook, NY, USA, 2020; Volume 33, pp. 596–608.
27. Miyato, T.; Maeda, S.-I.; Koyama, M.; Ishii, S. Virtual Adversarial Training: A regularization method for supervised and semi-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 1979–1993. [CrossRef]
28. Zhang, Z.; Zhao, Y.; Chang, M.-C.; Lin, C.; Liu, J. E4: Energy-efficient DNN inference for edge video analytics via early-exit and DVFS. *arXiv* **2025**, arXiv:2503.04865. [CrossRef]
29. Howard, A.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.-C.; Tan, M. Searching for MobileNetV3. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019*; IEEE: Piscataway, NJ, USA, 2019; pp. 1314–1324. [CrossRef]
30. Wang, C.-Y.; Bochkovskiy, A.; Liao, H.-Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696. [CrossRef]
31. Ratnayake, A.M.B.; Majid, M.S.; Yasin, H.; Naim, A.G.; Abas, P.E. Edge-Based Multi-Scale Predator Detection for Stingless Bee Protection Using Attention-Integrated YOLOv11. *Technologies* **2026**, *14*, 246. [CrossRef]
32. Wachowicz, A.; Pytlik, J.; Małysiak-Mrozek, B.; Tokarz, K.; Mrozek, D. Edge Computing in IoT-Enabled Honeybee Monitoring for the Detection of Varroa Destructor. *Int. J. Appl. Math. Comput. Sci.* **2022**, *32*, 355–369. [CrossRef]
33. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
34. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In *Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018*; IEEE: Piscataway, NJ, USA, 2018; pp. 7132–7141. [CrossRef]

35. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
36. Yu, F.; Koltun, V. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016), San Juan, Puerto Rico, 2–4 May 2016*; ICLR: San Juan, Puerto Rico, 2016; pp. 1–11. [CrossRef]
37. Hou, Q.; Zhou, D.; Feng, J. Coordinate attention for efficient mobile network design. In *Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021*; IEEE: Piscataway, NJ, USA, 2021; pp. 13633–13642. [CrossRef]
38. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018*; Springer: Cham, Switzerland, 2018; pp. 833–851. [CrossRef]
39. Gao, J.; Huang, Z.; Lei, Y.; Shan, H.; Wang, J.Z.; Wang, F.-Y.; Zhang, J. Deep rank-consistent pyramid model for enhanced crowd counting. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 299–312. [CrossRef]
40. Zhu, P.; Li, J.; Cao, B.; Hu, Q. Multi-task credible pseudo-label learning for semi-supervised crowd counting. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 10394–10406. [CrossRef]
41. Lu, H.; Cao, Z. TasselNetV2+: A Fast Implementation for High-Throughput Plant Counting From High-Resolution RGB Imagery. *Front. Plant Sci.* **2020**, *11*, 541960. [CrossRef] [PubMed]
42. Zhang, Z.; Li, Y.; Cao, Y.; Wang, Y.; Guo, X.; Hao, X. MTSC-Net: A Semi-Supervised Counting Network for Estimating the Number of Slash Pine New Shoots. *Plant Phenomics* **2024**, *6*, 0228. [CrossRef]