Hybrid Convolutional Network for Object Detection and Multi-Class Counting

Yuanlin Ning[1], Ying Yang[1(✉)], Zhenbo Li[1,2,3,4,5], Jianquan Li[1] and Ping Song[1]

[1] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
[2] Key Laboratory of Agricultural Information Acquisition Technology, Ministry of Agriculture and Rural Affairs, Beijing 100083, China
[3] Key Laboratory of Smart Farming for Aquatic Animal and Livestock, Ministry of Agriculture and Rural Affairs, Beijing 100083, China
[4] Precision Agricultural Technology Integration Research Base (Animal Husbandry), Ministry of Agriculture and Rural Affairs, Beijing 100083, China
[5] National Innovation Center for Digital Fishery, Ministry of Agriculture and Rural Affairs, Beijing 100083, China
hbxtyy@126.com

Abstract. In this work, we introduce a novel Hybrid Convolutional Network designed for efficient object detection and multi-class counting in varied applications such as aerial photography and surveillance. Leveraging the strengths of hybrid networks, our model facilitates the simultaneous execution of detection and counting tasks by sharing common network structures, thereby accelerating the image analysis process and enhancing feature generalization. We propose a novel Density-Aware Non-Maximum Suppression algorithm that adaptively adjusts the Intersection over Union (IoU) threshold according to object density, ensuring robust detection performance in both dense and sparse scenes. Additionally, we introduce a Region Suppression Module that leverages detection outcomes to minimize noise in density maps, further improving counting accuracy. Through comprehensive experiments, our approach demonstrates state-of-the-art performance in counting tasks and competitive accuracy in detection tasks across various datasets, while maintaining high processing speed.

Keywords: Object Detection, Object Counting, Multi-Task Learning.

1 Introduction

Object detection and counting play important roles in computer vision tasks and have gained increasing attention in a wide range of applications, including aerial photography and surveillance. Most detection methods achieve high accuracy in detection but have low accuracy in counting. In contrast, most counting methods focus on counting accuracy without precise localization of each individual. Therefore, it is very challenging to take both requirements into account in real-world scenarios.

Most methods handle these tasks separately. In terms of detection, for instance, Wang et al. proposed the multi-scale information prediction network MA-YOLO to improve the detection ability on unmanned aerial vehicles-captured scenarios [1]. Meethal et al. proposed the efficient Cascaded Zoom-in detector that re-purposes the detector itself for density-guided training and inference [2]. Although the above methods can achieve great detection accuracy, they often yield unsatisfactory counting results in cases of dense scenes and occlusions. In terms of counting, contemporary advanced counting methods predominantly rely on density map estimation. For instance, Fu et al. proposed MSCNet [3], a novel multi-scale dilated convolution channel-aware deep network for vehicle counting. Elharrouss et al. proposed a dilated and scaled neural networks for feature extraction and density crowd estimation [4]. Despite the excellent performance these methods achieve, counting methods based on density map estimation can't accurately localize the object.

Naturally, to achieve both satisfactory detection and counting accuracies, one apparent approach is to deploy separate models for object detection and counting. Nevertheless, using the two models to process this task separately would be more computationally expensive. A hybrid network is more suitable and efficient in this situation because 1) it can share common network structures to accelerate the image analysis process, 2) it generally has performance advantages as it learns more generalized features than a single-task network under the same dataset [5]. Therefore, it is essential to explore multi-task approaches that can simultaneously perform detection and counting tasks.

This paper proposes a novel approach for simultaneous object detection and multi-class counting. The overall workflow of the proposed method is shown in Fig. 1. Firstly, we propose a hybrid network that concurrently detects objects and estimates counts, comprising several key modules: a shared Backbone Network and Feature Fusion Module for feature extraction and fusion, followed by separate detection and density map estimation branches. For the predicted detection boxes, we propose a novel Density-Aware Non-Maximum Suppression algorithm that adaptively adjusts the Intersection over Union (IoU) threshold according to object density, ensuring robust detection performance in both dense and sparse scenes. For the predicted density maps, we propose a Region Suppression Module, which leverages predicted detection boxes to assist in producing high-quality density maps and effectively reducing noise. The main contributions of this paper are as follows:

1. We propose a hybrid convolutional network architecture that efficiently combines object detection and multi-class counting tasks. This integration allows for simultaneous processing, which not only accelerates the analysis of images but also enhances the generalization capability of the network across various tasks.
2. We introduce a novel Density-Aware NMS algorithm that dynamically adjusts the IoU threshold based on the local density of objects. This adaptive approach ensure robust detection performance in both dense and sparse scenes.
3. We propose a Region Suppression Module that utilizes detection results to refine density maps. This module effectively reduces noise in the density maps, leading to more accurate and reliable counting outcomes.

4. We formulate an Adaptive-Weight Joint Loss function that enables the end-to-end training of our hybrid network. This approach optimally balances the learning between detection and counting tasks, which results in improved overall performance of the network.

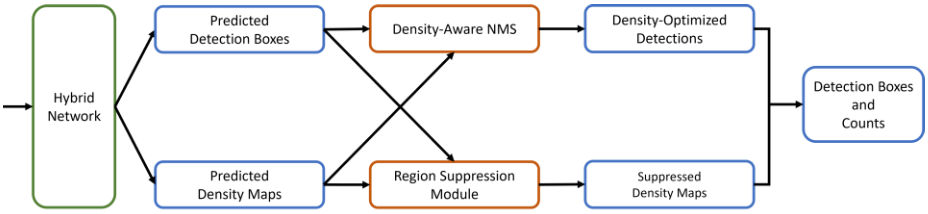

Hybrid Network
Predicted Detection Boxes
Predicted Density Maps
Density-Aware NMS
Region Suppression Module
Density-Optimized Detections
Suppressed Density Maps
Detection Boxes and Counts

Fig. 1. The overall workflow of the proposed method.

2 Related work

2.1 Object Detection

With the rapid development of deep learning, significant progress has been made in the field of object detection. Current mainstream object detection algorithms can be divided into two-stage detectors and one-stage detectors.

Two-stage detectors complete the task in two steps: first, a set of proposals is obtained from the feature maps, and then features in these proposals are used to locate and classify the objects. Examples of two-stage methods include R-CNN [6], Fast R-CNN [7] and Faster R-CNN [8].

Single-stage detectors treat object detection as a regression problem and use a unified deep neural network for feature extraction, object classification and bounding box regression, achieving end-to-end reasoning. SSD [9] and YOLO-series [10-17] are milestones in one-stage detection methods.

2.2 Object Counting

Contemporary advanced counting techniques predominantly rely on density map estimation. However, most of these methods were initially developed for tasks involving the counting of single-class objects. For instance, MCNN [18] employs a multi-column architecture designed to capture features at various scales, making it highly effective for scenes with varying object sizes. CSRNet [19] leverages a deep learning model that incorporates dilated convolutions to extend the field of view of filters, enabling effective feature extraction in crowded scenes.

Transitioning to multi-class object counting, the landscape is markedly less explored. A pivotal study by Wei et al. introduces the Dilated-Scale-Aware Category-Attention ConvNet (DSACA), a groundbreaking framework specifically devised for multi-class counting tasks [20]. This innovative approach not only tackles the inherent challenge of multi-class counting but also introduces the Category-Attention Module (CAM). CAM is ingeniously designed to mitigate the inter-class interference often encountered in density maps, marking a significant stride towards resolving one of the key complexities in multi-class object counting.

3 Approach

This section offers a detailed exposition of various aspects, including label generation, network architecture, postprocessing and loss function design.

3.1 Density Map Label Generation for Multi-Task Model

Most counting methods use point labels to represent objects in images, applying a 2-D Gaussian kernel to create a density map. However, this approach struggles with determining the appropriate kernel size and variance for varied object sizes and scales. In the proposed detection-counting multi-task framework, we utilize bounding box annotations to generate density maps that more accurately reflect the size and scale of objects within an image. Moreover, to support multi-class counting, our method introduces multi-class density maps to distinguish between different types of objects.

To generate the multi-class density map label $\mathrm{M}^{\mathrm{label}} = \{\mathrm{M}_1^{\mathrm{label}}, \dots, \mathrm{M}_{\mathrm{C}}^{\mathrm{label}}\}$, we position the center of each Gaussian kernel $G(x, y)$ at the object's centroid within its bounding box and sum up the contributions of all kernels across the image. This process can be formalized as:

$$M_c^{label}(x, y) = \sum_{i=1}^{N_c} G(x - x_i, y - y_i) \tag{1}$$

Here, N_c represents the number of objects belonging to category c in the image, and (x_i, y_i) denotes the centroid coordinates of the i-th object's bounding box. This formulation ensures that each pixel value of M_c^{label} represents the estimated density of objects with class c at that location, with higher values indicating greater likelihoods of object presence.

For each object within an image, represented by its bounding box, we compute a 2-D Gaussian kernel $G(x, y)$ as follows:

$$G(x, y) = \gamma G(x) G(y) \tag{2}$$

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma_x} e^{-\frac{(x-x_i)^2}{2\sigma_x^2}}, \; -\left\lfloor \frac{k_x}{2} \right\rfloor \le x \le \left\lfloor \frac{k_x}{2} \right\rfloor \tag{3}$$

$$G(y) = \frac{1}{\sqrt{2\pi}\sigma_y} e^{-\frac{(y-y_i)^2}{2\sigma_y^2}}, \; -\left\lfloor \frac{k_y}{2} \right\rfloor \le x \le \left\lfloor \frac{k_y}{2} \right\rfloor \tag{4}$$

Here, $G(x)$ and $G(y)$ represent Gaussian functions along the x and y axes, respectively. γ represents the amplification factor used to amplify the difference in values in the density map, and experiments have shown that it can improve performance. k_x and k_y represent the kernel size in x and y axes, respectively.

In point annotation based methods, the Gaussian kernel sizes k_x and k_y can only be estimated. As a multitasking network, we take the width w_i and height h_i of each object's bounding box as the corresponding Gaussian kernel sizes, as follows:

$$k_x = w_i \tag{5}$$

$$k_y = h_i \tag{6}$$

As for the variance σ of Gaussian function, we follow the traditional approach as follows [18]:

$$\sigma_x = \delta k_x \tag{7}$$

$$\sigma_y = \delta k_y \tag{8}$$

Here, δ is a hyperparameter.

3.2 Hybrid Network

As shown in Fig. 1, the architecture of our network is conceptually straightforward. Initially, it comprises a shared Backbone and a Feature Fusion Module to extract and fuse features. Subsequently, the output of the Feature Fusion Module is fed into the Detection Branch and Multi-Class Density Map Estimation Branch to fulfill their respective tasks. The Detection Branch employs an anchor-free multi-scale detection approach, while the Multi-Class Density Map Estimation Branch is designed to support multi-class counting. All layers are specifically designed to be lightweight and efficient, thereby gaining fast processing speed.

Backbone and Feature Fusion Module. The primary function of the backbone network is to extract features from images. However, due to perspective effects, the scale of objects within a scene can vary significantly, and directly utilizing deep features generated by the backbone network may lead to the loss of features for smaller objects. To address this challenge, various methods have been proposed, such as FPN [21], among others. Considering a balance between accuracy and lightweight design, we employ CSPDarknet [17] and PANet [22] as our Backbone and Feature Fusion Module, respectively. We select the outputs of P3, P4, and P5 (corresponding to 8x, 16x, and 32x downsampling relative to the original image) as the inputs for subsequent modules.

Detection Branch. As shown in Fig. 2, in the Detection Branch, we utilize anchor-free and decoupled heads currently used in advanced object detection techniques such as YOLOX [23], YOLOv6 [15], YOLOv7 [16], and YOLOv8 [17]. Specifically, the detection branch adopts a series of convolutional layers with a kernel size of 3, followed by batch normalization and SiLU activation functions. In the end, a convolutional layer with a kernel size of 1 is applied. The prediction is split into two separate tasks: bounding box regression and class prediction. This allows the network to adjust its weights more effectively for each specific task.

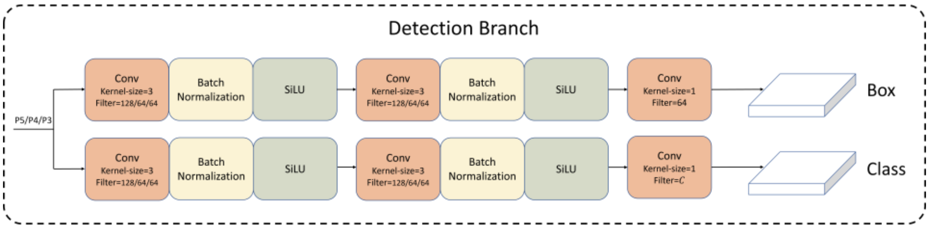


Fig. 2. The architecture of Detection Branch. 'Filter=$a/b/c$' denotes the filter sizes of the convolution layers, corresponding to the inputs P5, P4, and P3, respectively.

Multi-Class Density Map Estimation Branch. As depicted in Fig. 3, our Multi-Class Density Map Estimation Branch is strategically designed for efficiency. Initially, we opt for P3 (which entails an 8x downsampling ratio relative to the original image size) as the foundational input. To augment the network's receptive field, dilated convolutions are strategically utilized within the counting head. With inference speed as a critical consideration, we consciously abstain from employing deconvolution or interpolation techniques for feature map enlargement within this branch. Furthermore, to facilitate the prediction of distinct density maps for each category, the dimensionality of the output layer is meticulously aligned with the number of classes.

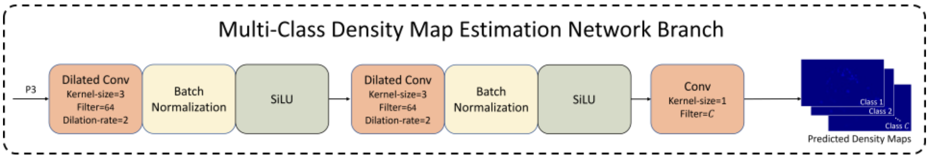


Fig. 3. The architecture of Multi-Class Density Map Estimation Branch.

3.3 Density-Aware NMS

In object detection frameworks, Non-Maximum Suppression (NMS) is crucial for reducing redundancy among bounding boxes, aiming to represent each detected object with a single, most accurate box. Traditional NMS methods (named Greedy-NMS) apply a fixed Intersection over Union (IoU) threshold to filter out overlapping boxes. However, this approach can be suboptimal in varying object density scenarios, where a single threshold may not suit both dense and sparse regions.

To address this challenge, Liu et al. introduced an Adaptive NMS [24]. First, it introduced a sub-network, taking the objectness predictions, bounding box predictions and conv features as input, to estimate the density score of each box. Then, during the NMS process, it adaptively adjusts the IoU threshold based on the density score.

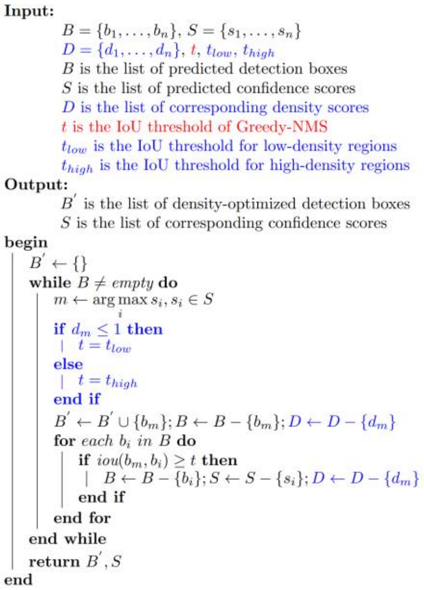
This methodology has inspired us to optimize the NMS process using the output of Multi-Class Density Map Estimation Branch. In Adaptive NMS, the density score is defined as the maximum bounding box IoU with other objects within the ground truth set. As a hybrid network, we can more intuitively measure the density score by the Multi-Class Density Map Estimation Branch. Specifically, for a predicted box b_i

belonging to category c, we define the corresponding density score d_i as the sum of values within the area of the box on a density map M_c belong to class c. The formula is as follows:

$$d_i = \sum_{(x,y)\in b_i} M_c(x, y) \tag{9}$$

This density score intuitively reflects the local density within the area of each box. In the density map, if objects do not overlap, the sum of the area within the density map for each box should equal 1 (This is determined by the properties of Gaussian kernels introduced by section. 3.1), that is, $d_i = 1$. Therefore, a sum greater than 1 suggests a high probability of overlap among objects.

Our method, named Density-Aware NMS, as outlined in the pseudo-code in Fig. 4. The cornerstone of our methodology is the adjustment of the IoU threshold for each box, contingent on its density score. In regions of low density (density score ≤ 1), indicative of minimal object overlap, we adopt a lower IoU threshold t_{low} to diminish box redundancy whilst maintaining detection precision. Conversely, in areas of high density (density score > 1), where object overlap is more probable, we impose a higher IoU threshold t_{high} to accommodate the closeness of valid detections.



```
Input:
        $B = \{b_1, \dots, b_n\}$, $S = \{s_1, \dots, s_n\}$
        $D = \{d_1, \dots, d_n\}$, $t$, $t_{low}$, $t_{high}$
        $B$ is the list of predicted detection boxes
        $S$ is the list of predicted confidence scores
        $D$ is the list of corresponding density scores
        $t$ is the IoU threshold of Greedy-NMS
        $t_{low}$ is the IoU threshold for low-density regions
        $t_{high}$ is the IoU threshold for high-density regions
Output:
        $B'$ is the list of density-optimized detection boxes
        $S$ is the list of corresponding confidence scores
begin
    $B' \leftarrow \{\}$
    while $B \neq empty$ do
        $m \leftarrow \arg\max_i s_i, s_i \in S$
        if $d_m \leq 1$ then
            $t = t_{low}$
        else
            $t = t_{high}$
        end if
        $B' \leftarrow B' \cup \{b_m\}; B \leftarrow B - \{b_m\}; D \leftarrow D - \{d_m\}$
        for $each\ b_i\ in\ B$ do
            if $iou(b_m, b_i) \geq t$ then
                $B \leftarrow B - \{b_i\}; S \leftarrow S - \{s_i\}; D \leftarrow D - \{d_m\}$
            end if
        end for
    end while
    return $B', S$
end
```

Fig. 4. The pseudo-codes of Greedy-NMS and Density-Aware NMS. Text highlighted in red represents the Greedy-NMS, while text highlighted in blue pertains to the Density-Aware NMS proposed in this paper.

3.4 Region Suppression Module

While Density Map Estimation Branch is capable of outputting count results for different classes, we observed that density maps for specific class are easily affected by other classes and backgrounds. Inspired by RoI (Region of Interest) of Faster R-CNN, we use the detections produced by the detection branch to generate an interesting region for each class, and then suppress non-interesting region, thereby reducing noise. Specifically, as shown in Fig. 5, first, a confidence threshold τ_{count} is employed for predicted detections B to produce interesting detections $B^{interest}$. Then, the interesting region, denoted as $\mathcal{M}$, is generated from interesting detections $B^{interest}$, which assigns a value of 1 to pixels within the boxes and 0 otherwise:

$$\mathcal{M}_c(x,y) = \begin{cases} 1 & if\,(x,y) \in B^{interest} \\ 0 & otherwise \end{cases} \tag{10}$$

Here, x and y are the coordinates in the density maps, and c represents the specific class of the object. The suppressed density map M' is obtained by Hadamard Product of the original density map M with the interesting region $\mathcal{M}$, filtering out the areas outside the region of interests and reducing noise:

$$M_c^{'}(x,y) = M_c(x,y) \odot \mathcal{M}_c(x,y) \tag{11}$$

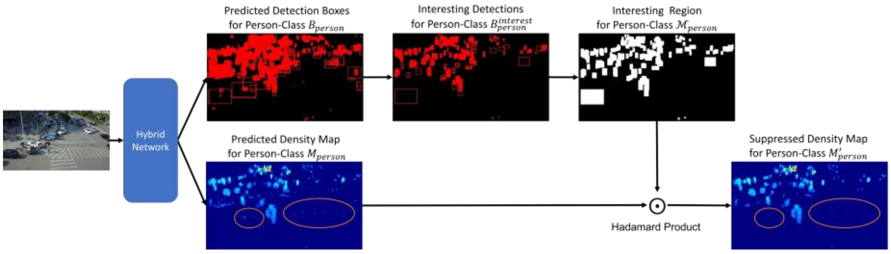

Predicted Detection Boxes for Person-Class B_{person}
Interesting Detections for Person-Class $B_{person}^{interest}$
Interesting Region for Person-Class $\mathcal{M}_{person}$
Hybrid Network
Predicted Density Map for Person-Class M_{person}
Suppressed Density Map for Person-Class M'_{person}
Hadamard Product

Fig. 5. Detailed workflow of the Region Suppression Module. The noise is indicated within the orange oval.

3.5 Loss Function

We introduce an adaptive-weight joint loss function to train the hybrid network in an end-to-end fashion. Firstly, we define the respective losses L_{det} and L_{count} for the detection task and counting task. Similar to other detection networks, the detection loss L_{det} is the sum of box loss L_{box} and classification loss L_{cls}, as follows:

$$L_{det} = L_{box} + L_{cls} \tag{12}$$

Here, the box loss L_{box} use the Complete Intersection over Union (CIoU) Loss [25] and Distribution Focal Loss [26] to minimize the difference between the predicted box and the ground truth bounding box. The classification loss L_{cls} use the binary cross entropy with logits loss, which combines a sigmoid function with the binary cross entropy loss.

For quantifying the counting loss, denoted as L_{count}, we employ the mean squared error (MSE) loss function. This choice is aimed at minimizing the discrepancy between the predicted density maps M and the density map labels M^{label} (the density map label generation method is detailed in Section. 3.1). Mathematically, the counting loss L_{count} is defined as follows:

$$L_{count} = \sum_{c=1}^{C} \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} \left(M_c(x,y) - M_c^{label}(x,y)\right)^2 \tag{13}$$

Here, C represents the number of classes, while W and H denote the width and height of the density map, respectively. $M_c(x,y)$ and $M_c^{label}(x,y)$ represent the predicted value and label at (x,y) in the c-axis channel.

To find a common representation for both detection and counting tasks within network layers, it is necessary to amalgamate the loss functions of multiple tasks through weighted combination. However, in the initial phases of training, we observe that the counting loss significantly outweighs the detection loss, thereby dominating the learning process. Subsequently, the counting loss rapidly diminishes, leading to a phase where detection loss becomes the predominant loss in the latter stages of training. This phenomenon makes the training process unstable. To circumvent this issue, we employ a multi-task loss function based on maximizing the Gaussian likelihood with homoscedastic uncertainty as suggested by Liebel et al. [27]. The joint loss L is defined as follows:

$$L = \frac{1}{2 \cdot p_{det}^2} L_{det} + \frac{1}{2 \cdot p_{count}^2} L_{count} + \ln(1 + p_{det}^2) + \ln(1 + p_{count}^2) \tag{14}$$

Here, p_{det} and p_{count} represent the learnable parameters designed to equilibrate the detection and counting losses, respectively. Since the network's optimization goal is to minimize the overall loss L, both $\frac{1}{2 \cdot p_{det}^2}$ and $\frac{1}{2 \cdot p_{count}^2}$ terms would prefer p_{det} and p_{count} to be as large as possible. To prevent degeneration, the $\ln(1 + p_{det}^2)$ and $\ln(1 + p_{count}^2)$ terms would prefer p_{det} and p_{count} to be as small as possible. When either the detection or counting loss is relatively high, the corresponding p_{det} or p_{count} will assume a larger value, thereby minimizing the overall loss. This mechanism is crucial for the optimization process, ensuring a balanced contribution from both detection and counting losses towards the overall learning objective.

4 Results and discussion

In this section, the effectiveness of our proposed approach is evaluated. First, we present details of our implementation. Then, we introduce the evaluation metrics and the benchmark datasets. After that, we discuss the counting and detection results and compare them to exist state-of-the-art methods. Finally, we conduct an ablation study.

4.1 Implementation Details

We use SGD optimizer with a learning rate of 0.01 and a momentum of 0.9. The δ and amplification factor γ for density map generation was set to 1 and 100, respectively. The IoU threshold t_{low} and t_{high} was set to 0.4 and 0.6, respectively. The confidence threshold τ_{count} for the Region Suppression Module was set to 0.001. The image resolution is uniformly resized to 1280×1280 pixels. All speed benchmarks are performed on an NVIDIA 2080Ti GPU.

4.2 Evaluation Metrics

To facilitate easy comparison, various well-accepted and widely used evaluation methods are employed. For object detection, we use mAP@0.5 and mAP@0.5:0.95 to evaluate our model. For multi-class counting, the mean absolute error (MAE) is used to evaluate our model. MAE_c is denoted as MAE of the c-th class in the N test images as:

$$MAE_c = \frac{1}{N}\sum_{i=1}^{N}\left|\sum_{x=0}^{W-1}\sum_{y=0}^{H-1} M_c(x,y) - \sum_{x=0}^{W-1}\sum_{y=0}^{H-1} M_c^{label}(x,y)\right| \tag{15}$$

Here, N represent the number of test images; W and H denote the width and height of the density map, respectively.

$M_c(x,y)$ and $M_c^{label}(x,y)$ represent the predicted value and label at (x,y) in the c-axis channel.

Furthermore, $mMAE$ for all C classes and N test images are defined as:

$$mMAE = \sum_{c=1}^{C} MAE_c$$

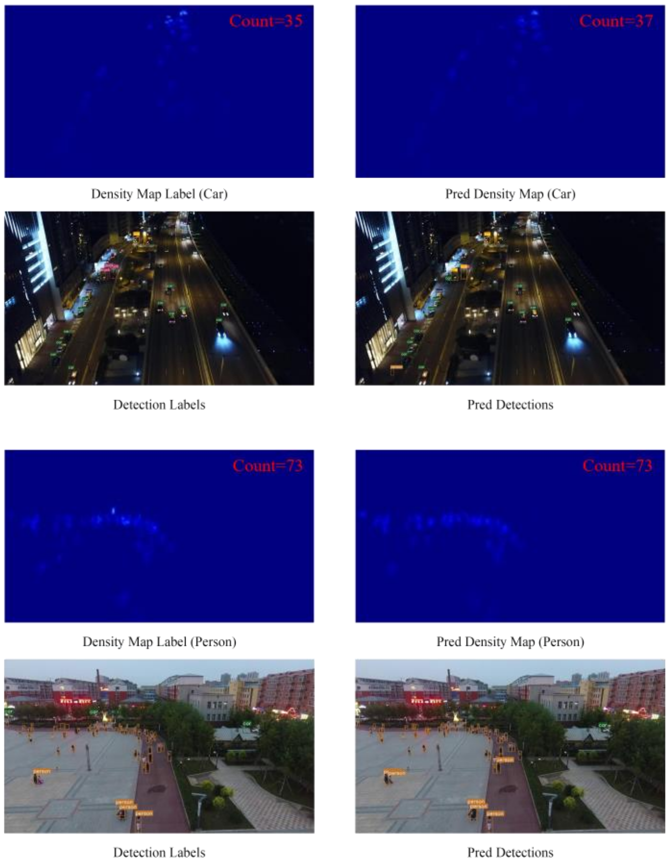
4.3 Dataset

We utilize the Visdrone-Det2019 dataset as our evaluation dataset. Visdrone-Det2019 dataset contains 10,209 static images (6,471 for training, 548 for validation and 3,190 for testing) captured by drone platforms in various locations at different heights, featuring ten classes (i.e., pedestrian, person, car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle). In line with other work [20], the pedestrian category has been merged into the person category, and the awning-tricycle category into the tricycle category.

4.4 Counting Results

We conducted comparative experiments between our proposed model and other single class counting models and multi-class counting models. To facilitate equitable comparisons, we adapted the output channels of prominent single-class models—namely, MCNN [18], SANet [28], CSRNet [19], BL [29], and CAN [30]—to accommodate multi-class object counting tasks. This adjustment aligns with the methodology employed in DSACA [20]. Additionally, we also illustrate the quality of the predicted density maps in Fig. 6.

Table 1 shows a comparison of counting accuracy between our model and other mainstream models. The proposed model achieves the best accuracy. Specially, the proposed model reduces 1.10 in $mMAE$, 1.36 in $MAE_{bicycle}$, 0.49 in MAE_{car}, 0.51 in MAE_{van}, 0.05 in MAE_{truck}, 2.21 in $MAE_{tricycle}$ and 6.6 in MAE_{motor} compared with DSACA [20].



Density Map Label (Car) Pred Density Map (Car)

Detection Labels Pred Detections

Density Map Label (Person) Pred Density Map (Person)

Detection Labels Pred Detections

Fig. 6. Visualizing results. For clarity of representation, in the detection visualization results, we have only plotted the labels of large detection boxes. In the density map visualization results, we have only plotted one primary category out of all categories. The predicted count has been rounded.

Table 1. Counting results. The bold and underline fonts respectively represent the first and second place.

Method	FPS	Mean MAE	Person MAE	Bicycle MAE	Car MAE	Van MAE	Truck MAE	Tricycle MAE	Bus MAE	Motor MAE
MCNN	66.20	5.66	12.27	2.35	17.89	2.82	1.34	2.93	0.43	5.26
SANet	18.15	7.54	25.48	2.38	15.27	3.61	1.37	2.90	**0.42**	8.92
CSRNet	20.42	4.59	9.10	2.49	8.50	5.96	1.83	2.82	0.79	5.27
BL	21.76	5.46	11.88	2.84	11.49	6.22	2.83	2.88	0.78	4.74
CAN	19.51	6.86	9.14	6.67	8.77	8.88	8.75	5.99	2.23	4.48
DSACA	11.68	3.43	**5.04**	2.35	3.98	2.54	1.32	2.88	**0.42**	8.90
Ours	**106.38**	**2.33**	6.73	**0.99**	**3.49**	**2.03**	**1.27**	**0.67**	1.17	**2.30**

4.5 Detection Results

Table 2 provides a comparison of our model against other mainstream detection models in terms of detection performance and we also illustrate the quality of predict detections in Fig. 6.

The results demonstrate that our model exhibits strong competitiveness in detection metrics. Specifically, our model achieves 0.580 in mAP@0.5 and 0.387 in mAP@0.5:0.95 with 106.38 FPS. Compared to YOLOv8s, our model shows an improvement of 2.2% in mAP@0.5 and 1.5% in mAP@0.5:0.95. Compared to the YOLOv8m model, the accuracy of the model we proposed is comparable, but it boasts an FPS improvement of nearly 200%.

Table 2. Detection results. The bold and underline fonts respectively represent the first and second place.

Method	FPS	Mean		Person		Bicycle		Car		Van		Truck		Tricycle		Bus		Motor	
		mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95
YOLOv5s	**135.13**	0.522	0.333	0.510	0.229	0.257	0.131	0.819	0.565	0.469	0.352	0.536	0.371	0.449	0.272	0.685	0.528	0.450	0.218
YOLOv8s	114.94	0.558	0.372	0.557	0.269	0.300	0.165	0.832	0.592	0.513	0.391	0.569	0.416	0.505	0.327	0.688	0.554	0.504	0.261
YOLOv8m	53.47	**0.582**	**0.389**	**0.586**	**0.285**	0.318	0.179	0.835	0.598	0.514	0.392	**0.616**	**0.455**	**0.533**	**0.344**	**0.716**	**0.579**	**0.536**	**0.279**
Ours	106.38	0.580	0.387	0.577	0.282	**0.325**	**0.185**	**0.845**	**0.602**	**0.539**	**0.411**	0.590	0.432	0.523	0.339	0.714	0.569	0.528	0.277

4.6 Ablation Study

In this section, we present an ablation study to evaluate various components of our model. Initially, we investigate the influence of the Adaptive-Weight Joint Loss on both detection and counting accuracy. Furthermore, we examine the effects of several counting improvements, including the Density Map Generation Method for Multi Task Model and Region Suppression Module. Lastly, we assess the impact of a detection improvement (Density-Aware NMS) on the detection accuracy.

Adaptive-Weight Joint Loss. The concept of Adaptive-Weight Joint Loss is instrumental in the training phase, serving to dynamically adjust the weights allocated to various loss components. As evidenced by the data presented in Table 3, the implementation of Adaptive-Weight Joint Loss has conferred notable benefits on our model. Specifically, it has been a diminution in mMAE by 0.19. Additionally, it has facilitated an improvement in the metric of mAP@0.5 by 5.1% and mAP@0.5:0.95 by 4.9%, further underscoring the efficacy of the Adaptive-Weight Joint Loss approach in enhancing model accuracy.

Table 3. Ablation study about Adaptive-Weight Joint Loss. The bold font represents the first place.

Adaptive Weight Joint Loss	Mean			Person			Bicycle			Car			Van			Truck			Tricycle			Bus			Motor		
	MAE	mAP 0.5	mAP 0.5:0.95	MAE	mAP 0.5	mAP 0.5:0.95	MAE	mAP 0.5	mAP 0.5:0.95	MAE	mAP 0.5	mAP 0.5:0.95	MAE	mAP 0.5	mAP 0.5:0.95	MAE	mAP 0.5	mAP 0.5:0.95	MAE	mAP 0.5	mAP 0.5:0.95	MAE	mAP 0.5	mAP 0.5:0.95	MAE	mAP 0.5	mAP 0.5:0.95
	3.85	0.525	0.336	6.94	0.528	0.242	1.94	0.289	0.150	5.90	0.806	0.560	**3.13**	0.471	0.346	2.15	0.511	0.359	2.40	0.455	0.284	5.34	0.671	0.519	2.99	0.469	0.230
√	**3.66**	**0.576**	**0.385**	**6.59**	**0.565**	**0.275**	**1.48**	**0.321**	**0.183**	**4.00**	**0.844**	**0.601**	5.73	**0.538**	**0.411**	**2.01**	**0.588**	**0.430**	**2.24**	**0.519**	**0.337**	**4.40**	**0.712**	**0.567**	**2.86**	**0.522**	**0.273**

Counting Improvements. In the object counting task, we introduce two main innovations: the Density Map Generation Method for the Multi-Task Model and the Region Suppression Module. The Density Map Generation Method for the Multi-Task Model can adjust the kernel size and variance in the Gaussian function, tailored to the size information of the objects. The Region Suppression Module is to suppress the noise of predicted density maps. As indicated in Table 4, benefiting from the Density Map Generation Method for the Multi-Task Model and the Region Suppression Module, our model has achieved a reduction in mMAE by 1.33.

Table 4. Ablation study about counting. The bold and underline fonts respectively represent the first and second place.

Density Map Generation Method for Multi-Task Model	Region Suppression Module	Mean MAE	Person MAE	Bicycle MAE	Car MAE	Van MAE	Truck MAE	Tricycle MAE	Bus MAE	Motor MAE
		3.66	**6.59**	1.48	4.00	5.73	2.01	2.24	4.40	2.86
√		2.77	7.17	1.56	5.46	**1.98**	**0.92**	1.51	**0.44**	3.09
√	√	**2.33**	6.73	**0.99**	**3.49**	2.03	1.27	**0.67**	1.17	**2.30**

Table 5. Ablation study about detection. The bold fonts represent the first place.

Density Aware NMS	Mean		Person		Bicycle		Car		Van		Truck		Tricycle		Bus		Motor	
	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95	mAP 0.5	mAP 0.5:0.95
	0.576	0.385	0.565	0.275	0.321	0.183	0.844	0.601	0.538	**0.411**	0.588	0.430	0.519	0.337	0.712	0.567	0.522	0.273
√	**0.580**	**0.387**	**0.577**	**0.282**	**0.325**	**0.185**	**0.845**	**0.602**	**0.539**	**0.411**	**0.590**	**0.432**	**0.523**	**0.339**	**0.714**	**0.569**	**0.528**	**0.277**

Detection Improvement. We propose the Density-Aware NMS algorithm to dynamically adjust the IoU threshold based on the local density of objects. This adaptive approach ensures robust detection performance in both dense and sparse scenes. As indicated in Table 5, benefiting from the Density-Aware NMS, our model shows an improvement of 0.4% in mAP@0.5 and 0.2% in mAP@0.5:0.95.

5 Conclusion

In this paper, we introduce a hybrid network, which is capable of simultaneously handling the dual tasks of object detection and counting, with the ability to be trained end-to-end. Additionally, we propose a Density-Aware NMS algorithm that adaptively adjusts the IoU threshold based on the object density. Furthermore, we propose a Region Suppression Module capable of utilizing detections to diminish noise in density maps. Our model demonstrates exceptional performance on the challenging Visdrone-Det2019 datasets in both tasks. In our future research, we aim to explore more innovative methods to further integrate detection and counting tasks.

Acknowledgments. This study was funded by the National Key R&D Program of China (Grant No. 2021ZD0113805).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Wang, C., Wei, X., Jiang, X.: MA-YOLO: Multi-Scale Information Prediction Network Based on the Multi-Direction Weighted Pyramid for UAV Scene. In: 2023 International Joint Conference on Neural Networks (IJCNN), pp. 01-08. IEEE, (2023)
2. Meethal, A., Granger, E., Pedersoli, M.: Cascaded zoom-in detector for high resolution aerial images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2045-2054. (2023)
3. Fu, Q., Min, W., Li, C., Zhao, H., Cao, Y., Zhu, M.: MSCNet: Dense vehicle counting method based on multi-scale dilated convolution channel-aware deep network. GeoInformatica 28, 245-269 (2024)
4. Elharrouss, O., Almaadeed, N., Abualsaud, K., Al-Ali, A., Mohamed, A., Khattab, T., Al-Maadeed, S.: Drone-SCNet: Scaled cascade network for crowd counting on drone images. IEEE Transactions on Aerospace and Electronic Systems 57, 3988-4001 (2021)
5. Zhang, C., Yang, H., Ma, J., Chen, H.: An Efficient End-to-End Multitask Network Architecture for Defect Inspection. Sensors 22, 9845 (2022)
6. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587. (2014)
7. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp. 1440-1448. (2015)
8. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems 28, (2015)
9. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14, pp. 21-37. Springer, (2016)
10. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. (2016)
11. Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7263-7271. (2017)
12. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
13. Bochkovskiy, A., Wang, C.-Y., Liao, H.-Y.M.: Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934 (2020)
14. Jocher, G., https://github.com/ultralytics/yolov5, last accessed 2022-09-01
15. Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., Ke, Z., Li, Q., Cheng, M., Nie, W.: YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976 (2022)
16. 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 preprint arXiv:2207.02696 (2022)

17. Jocher, G., Chaurasia, A., Qiu, J., https://github.com/ultralytics/ultralytics, last accessed 2023-07-01
18. Zhang, Y., Zhou, D., Chen, S., Gao, S., Ma, Y.: Single-image crowd counting via multi-column convolutional neural network. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 589-597. (2016)
19. Li, Y., Zhang, X., Chen, D.: Csrnet: Dilated convolutional neural networks for understanding the highly congested scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1091-1100. (2018)
20. Xu, W., Liang, D., Zheng, Y., Xie, J., Ma, Z.: Dilated-scale-aware category-attention convnet for multi-class object counting. IEEE Signal Processing Letters 28, 1570-1574 (2021)
21. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2117-2125. (2017)
22. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J.: Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8759-8768. (2018)
23. Ge, Z., Liu, S., Wang, F., Li, Z., Sun, J.: Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430 (2021)
24. Liu, S., Huang, D., Wang, Y.: Adaptive nms: Refining pedestrian detection in a crowd. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 6459-6468. (2019)
25. Zheng, Z., Wang, P., Ren, D., Liu, W., Ye, R., Hu, Q., Zuo, W.: Enhancing geometric factors in model learning and inference for object detection and instance segmentation. IEEE Transactions on Cybernetics (2021)
26. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-IoU loss: Faster and better learning for bounding box regression. In: Proceedings of the AAAI conference on artificial intelligence, pp. 12993-13000. (2020)
27. Liebel, L., Körner, M.: Auxiliary tasks in multi-task learning. arXiv preprint arXiv:1805.06334 (2018)
28. Cao, X., Wang, Z., Zhao, Y., Su, F.: Scale aggregation network for accurate and efficient crowd counting. In: Proceedings of the European conference on computer vision (ECCV), pp. 734-750. (2018)
29. Ma, Z., Wei, X., Hong, X., Gong, Y.: Bayesian loss for crowd count estimation with point supervision. In: Proceedings of the IEEE/CVF international conference on computer vision, pp. 6142-6151. (2019)
30. Liu, W., Salzmann, M., Fua, P.: Context-aware crowd counting. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 5099-5108. (2019)