Yang *et al. EURASIP Journal on Image and Video Processing* (2020) 2020:29
https://doi.org/10.1186/s13640-020-00517-3

EURASIP Journal on Image
and Video Processing

RESEARCH Open Access

Anchor-free object detection with mask attention

He Yang, Beibei Fan* and Lingling Guo

* Correspondence: fanbeibei@shu.edu.cn
School of Mechatronic Engineering and Automation, Shanghai University, Shanghai 200072, China

Abstract

The anchor-free method based on key point detection has made great progress. However, the anchor-free method is too dependent on using a convolutional network to generate a rough heatmap. This is difficult to detect for objects with a large size variation and dense and overlapping objects. To solve this problem, first, we propose a mask attention mechanism for object detection methods and make full use of the advantages of the attention mechanism to improve the accuracy of network detection heatmap generation. Then, we designed an optimized fire model to reduce the size of the model. The fire model is an extension of grouped convolution. The fire model allows each group of convolutional network features to learn the same feature through purposeful grouping. In this paper, the mask attention mechanism uses object segmentation images to guide the generation of corner heatmaps. Our approach achieved an accuracy of 91.84% and a recall of 89.83% in the Tencent-100 K dataset. Compared with the popular object detection methods, the proposed method has advantages in model size and accuracy.

Keywords: Object detection, Anchor-free, Convolutional neural network, Attention mechanism

1 Introduction

In recent years, applications based on object detection technology have become more widespread [1]. Common applications are pedestrian detection [2], vehicle detection [3], image retrieval [4], and traffic sign detection [5]. This imposes higher requirements on the detection performance and size of the object detection method. As a typical method of object detection, anchor-based object detection can often be seen in common application scenarios. Anchor-based object detection methods have made great progress in location and recognition performance [6, 7]. These anchor-based methods regress the boundaries of objects by generating dense coordinates on the feature map. In order to get a higher quality coordinate box, non-maximum suppression method is used to filter to most of the overlap bounding boxes. However, computing anchor boxes and filter boxes requires a large amount of computing resources. At the same time, the pre-designed anchor box is not aligned with the real boundary of the object. As a result, anchor-based has defects in calculation and accuracy.

To solve the anchor box problem in the anchor box, the first step is to prioritize the limitations of manually setting the anchor box. Recently, many anchor box-free methods have emerged such as CornerNet and CenterNet [8, 9]. The early DenseBox [10] method directly used landmark heatmaps and face score map to predict human face. It used the combination of key points and heatmaps. The effect of box regression using only convolutional networks is not obvious, leading to reduced accuracy and recall. YOLO divides the feature map into 7 × 7 squares. Object detection is carried out in each square. Because the scale of the object varies too much, the network is difficult to learn. As a result, YOLO [6] mAP is low and recall is low. In order to enhance the detection performance of small objects, YOLO-v2 [11] and YOLO-v3 [12] join the anchor-boxed side. CornerNet transforms the position detection of an object into the detection of the key points of the top-left corner and the bottom-right corner of the boundary box. This method based on key point detection simplifies the process of object detection and greatly simplifies the task of the convolution network.

CornerNet simplifies the process of object detection by using key point detection technology. Because CornerNet is based on hourglass network [13], the hourglass network is redundantly calculated, which makes it difficult to train in the absence of computing resources. The optimized version of CornerNet, CornerNet-Squeeze [14], solves this problem to some extent. However, the detection performance of CornerNet-Squeeze is unsatisfactory due to the excessive pursuit of compression and simplification and the lack of consideration of global spatial information. CornerNet-Saccade [14] is designed to increase the network global attention to the image. Another optimization strategy uses a similar two-stage object detection method. The method guides the cropped image by the attention mechanism and uses the cropped image to detect to improve efficiency. This method is not contiguous in backpropagation, which is detrimental to the overall weight update.

Inspired by CornerNet-Squeeze and CornerNet-Saccade, we propose a Mask-CornerNet. In order to make the convolutional network pay attention to the region information of objects contained in the space, this paper proposes a mask module. The mask module is similar to the method of segmentation model. It is different from other object segmentation methods. The results of the mask module in this paper are fed back to the corner detection network. The mask module can constrain the network to pay more attention to the area containing the object. The recognition results of our method are shown in Fig. 1. The SGE module [15] gets inspiration from grouped convolution. It can greatly improve network performance by introducing few parameters, which is very helpful for enhancing channel information. We optimized the fire model with this idea and designed a new fire model.

In this paper, we optimized the design of CornerNet-Squeeze. While improving the detection accuracy, the calculation efficiency is maintained. It improves detection accuracy by adding a mask branch. Figure 6 shows some of the results of our method. Our method is compatible with speed and accuracy. The main contributions are as follows:

1. We propose a fire model module with fewer parameters and higher efficiency. In the fire model, we take advantage of the group convolution to fully exploit the local and global relationships of the convolution channel.

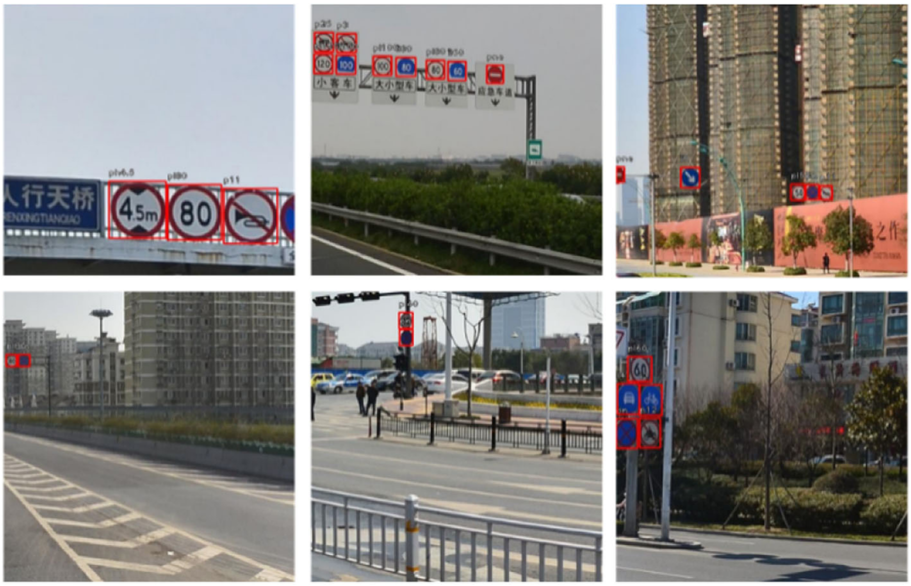

Fig. 1 Object detection results of some real traffic scenes. Our system works very well in a complex scene where objects are small and highly occluded

2. We propose to add a mask module. The mask module has the ability to segment objects. It can use the object segmentation feature map to assist in generating the corner heatmap, which effectively enhances the expression ability of the convolution feature.
3. Our approach takes advantage of the features of the hourglass network. We use the fusion method of context features to improve the ability to express features.

The rest of this paper will be organized as follows. Section 2 mainly analyzes the related work of current object detection. The main work description of our work is in Section 3. Section 4 shows the experimental results. Finally, we summarize the work and give our conclusion.

2 Related work

2.1 Anchor-based detectors

Since the advent of the R-CNN series of object detection methods based on convolutional networks, anchor-based detectors have become very popular. Today, there are many advanced object detection methods such as Faster-rcnn [7], SSD [16], FPN [17], RetinaNet [18], and Mask-rcnn [19]. These methods benefit from the clever design of the anchor, and the detection performance is steadily increasing. The R-CNN approach opens the era of anchor-based methods. R-FCN [20], FPN, and Mask-RCNN are typical RPN methods. At the same time, there are methods to improve the expression ability of model features by optimizing the shape and size of the convolution kernel, for example, atrous convolution [21], depthwise separable convolutions [22], and deformable convolutional [23]. Deformable convolutional networks are also an innovative deep learning optimization direction, breaking through the innovation direction of traditional convolutional network.

Recently, it is best to use multiple receptive field branches and take a weight-sharing approach to achieve the best results based on Anchor-based [24].

The anchor-based method constrains the regression results by proposing dense anchor points on the image. This method simplifies the learning task of convolutional networks, but makes the detection method less flexible. Thus, this results in a convolutional network predicting regression coordinate size limits, fixed shape, fixed aspect ratio, and small plasticity. The anchor box is always not aligned with the regression results, resulting in a small intersection over union (IOU) value of the box. The anchor aspect ratio, size, and shape in reality are highly uncertain. A manually set anchor cannot match it. In the training phase, the regression task needs to be performed by comparing each result with the IOU of the anchor. The time cost and space consumption are high. Based on the above reasons, this article uses anchor-free object detection method. This method has no fixed anchor constraints and can more flexibly match objects of various proportions.

2.2 Anchor-free detectors

Anchor-free method is another popular method, and YOLO has been considered as the representative of anchor-free object detection. YOLO method was to get rid of the R-CNN series of methods for two-stage inference. YOLO coordinates and classifies objects directly from the image. This method of directly returning to the frame coordinates without using an anchor point greatly improves the speed of object detection. However, this rough detection method results in low accuracy and inability to detect small objects. DenseBox is considered to be the earliest anchor-free method [10]. DenseBox uses five heatmaps to classify faces and two corner coordinates. Because of the defects of DenseBox in the overlapping processing of heatmaps, the recall rate of DenseBox in general object detection is low.

CornerNet removes the anchor box setting and uses the top-left corner and the bottom-right corner to represent the boundaries of a box [25]. Main task of CornerNet is to predict the position of two corners, and classified tasks are distributed in corner detection tasks. In order to improve the recall rate, corner pool is used to maximize each other at the two boundaries in order to improve the probability of the occurrence of objects. Inspired by Cornernet, CenterNet uses the midpoint of the object as the detection center [8]. It can detect the object by adding the boundary and size information of the object. FCOS [26] uses center-ness to suppress excessive and inappropriate box boundaries and uses multiple and classified objects to classify. ExtremeNet [27] uses standard key points to estimate four poles (top, left, bottom, right) and a center point of the network detection object. Compared with other anchor-free methods, CornerNet makes full use of the edge information of objects. In particular, the corner embedding vector makes the pair of corners clear.

2.3 Object segmentation

The performance of object segmentation methods based on convolutional networks is increasingly higher. FCN [28] directly uses a convolutional network to make binary predictions for each category. The FCN method uses a convolutional neural network to extract the semantic features of the image. This segmentation method gets rid of traditional color and shape features, thereby reducing the difficulty of image segmentation. Mask-rcnn directly adds the segmentation task to the object recognition task, which provides a new idea for object detection. After that, U-Net [29] and V-Net [30] encoder-decoder networks made full use of skip structures for feature fusion and made great progress in the fields of medicine and 3D detection. Our approach is different from the above. We use the results of the segmentation module to guide the convolution to generate new features.

3 The proposed method

Our approach is based on CornerNet-Squeeze. This method uses corner points to represent the bounding box of the object. Use the heat map to directly return coordinates. In order to eliminate the deviation of the coordinates, use offset to correct. By analyzing the network structure and overall method of CornerNet, it is found that the heat map is the most critical point of the whole method, especially the key position and the value of the heatmap value. But this feature is not really utilized in the entire network model. And the detection image of the corner point is completely dependent on the heat map. The accuracy of the corner heat map depends entirely on the training data and the feature extraction ability of the convolutional neural network. However, it is more difficult to completely rely on the convolutional network to generate an accurate corner heat map. Object segmentation can highlight the area containing the target. Inspired by this, this paper uses a masking module to enhance the expression of object. The main function of the mask module is to generate a complete segmented image based on the original image. The segmentation result image is fused with the internal feature map of the convolutional network to generate enhanced features.

3.1 Base framework

This section describes the overall framework and process of the proposed method. Figure 2 illustrates the overall framework of our approach. Our method uses two hourglass network structures in tandem, which effectively utilizes the contextual features of the image. Based on the symmetrical nature of the hourglass network, the backbone network is elegant. The hourglass network is followed by the mask module. Our mask model is more than just an additional separate module; it is a feature-enhanced

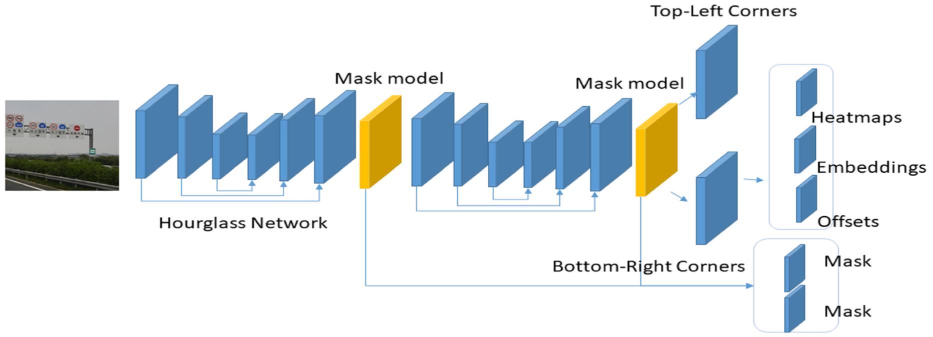

Fig. 2 Describe the proposed method. By stacking multiple hourglass network and mask modules to enhance the network extraction ability, the mask module can get the image segmentation result. The corner detection module has two parts, and the top left and bottom right corners of the object are obtained

function module. The mask module results are directly fed back to the original feature map to constrain the feature map.

The mask module focuses on the critical areas containing objects by constraining the convolutional network, thereby improving the detection accuracy of the network. In order to fully utilize and mine the contextual features of the feature map, we combine the outputs of the two hourglass network elements by element.

The Squeeze method can sufficiently reduce the number of channels in the convolutional network. Because the rough grouping method does not highlight the relationship between the channels, the relationship between the groups cannot be directly mined. The Spatial Grouping Enhancement (SGE) module can generate an attention factor for each spatial position in each convolutional semantic group [31]. Therefore, we adjust the importance of each channel group function through SGE so that each individual group can autonomously enhance its learned expression and suppress possible noise. With almost no additional learning parameters added, we optimized the fire module. The advantages of group convolution can reduce the computational and parametric advantages. By separately performing local and global feature similarity for each set of convolutions, the spatial distribution of semantic features is enhanced to achieve a clear division of labor for each channel.

The coordinate regression and classification of the network is realized by detecting the top-left and bottom-right. The final detection of the network includes two branches, top-left and bottom-right detection modules. Detection modules include heatmaps, offsets, and embeddings.

The heatmap contains C channels (C is the target category, no background category), and each channel is a binary mask indicating the angular position of the corresponding category. For each corner point, there is only one ground-truth, and the other positions are negative samples. During the training process, the model reduces the negative samples and sets a positive sample in the radius r region at each ground-truth corner. This is because the vertices falling within the radius r region can still generate a valid boundary of objects.

Offsets are a correction module that is specifically set to correct errors in the heatmap corners. Due to the presence of down-sampling, the heatmap size generated by the model is smaller than the original input image. After n times of downsampling, the position of the point (i, j) mapped onto the heatmap on the original image becomes ($\lfloor x/n \rfloor$, $\lfloor y/n \rfloor$). Re-mapping the points on the heatmaps to the original image has a quantization map error, resulting in an offset in the corner position of the map. This offset can seriously affect small object IOU calculations. The correction of the detection result is obtained by adding the heat map prediction result and the offset.

Embedding vectors are the key to corner detection. By encoding the corner points, the relationship of the embedded vectors in the top-left and bottom-right corners is obtained. The distance of the embedding vector of two corner points is used in the article to determine whether it is a pair of corner points of the same object.

3.2 Fire module

The use of the group convolution method can reduce the use of parameters, but cannot improve the feature representation of the convolutional network. In order to enhance the feature representation of the model, we absorbed the inspiration of the SEG module

[31]. The fire module is shown in Fig. 3. Each SGE module increases the number of parameters by about 2 times the number of groups, and the number of groups is usually 32 or 64.

The fire module groups feature on channels. Group convolution divides a channel into multiple sub-functions to represent different semantics. For a particular set of semantics, it is reasonable and beneficial to generate corresponding semantic features at the correct spatial location of the original image. In the fire module of this paper, first, squeeze the internal convolution feature maps of each group to obtain the global semantic parameters of each feature map, such as Eq. (1). Multiply the semantic parameters and the original features element-by-element, to obtain the feature map c_i of each channel. Then normalize the feature map c_i in each group. The normalization of c_i introduces only two learning parameters, allowing SEG to automatically normalize. The activation function is used to assign weight to feature a_i, and finally, a_i is multiplied element-by-element with the original feature.

First, the features are grouped. Spatial averaging function F_{gp} can get the global semantic feature *g*. This allows you to quickly get the global semantics of each grouping.

$$\mathrm{g} = F_{gp}(x) = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{1}$$

In the convolution integral group, the global feature of each channel is multiplied by the original feature point to obtain the initial attention feature c_i.

$$\mathrm{c}_i = \mathrm{g}\cdot x_i \tag{2}$$

The mean μ_c within the group is subtracted from each group and divided by the variance σ_c within the group. The two scaling offset parameters allow the normalize operation to be restored, then the sigmoid to get the final attention mask and scale the feature at each location in the original feature group.

$$\widehat{\mathrm{c}_i} = \frac{\mathrm{c}_i - \mu_c}{\sigma_c},\quad \mu_c = \frac{1}{m}\sum\nolimits_j^m c_j,\quad \sigma_c^2 = \frac{1}{m}\sum\nolimits_j^m \left(c_j - \mu_c\right)^2 \tag{3}$$

Similar to BN, in order to enhance standardization versatility, add two additional parameters (*λ*, *β*) to represent the standard scale and offset. This is the two sets of non-convolution parameters that need to be learned in the fire model.

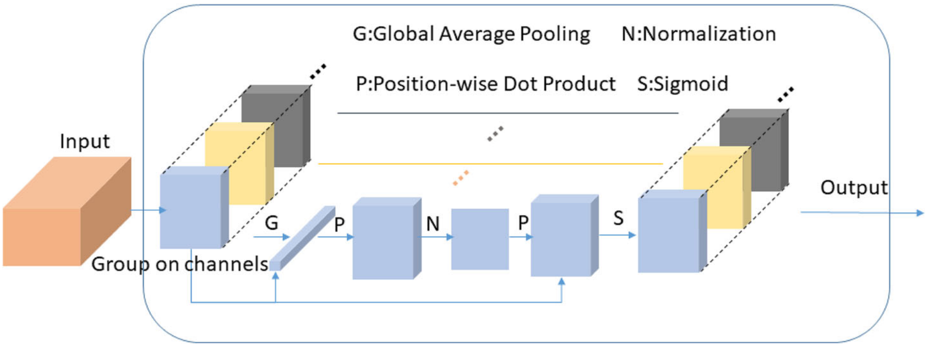

Fig. 3 Diagram our improved fire module, which combines global and local features by grouping for incoming feature maps

$$\mathrm{a}_i = \lambda \widehat{\mathrm{c}_i} + \beta \tag{4}$$

In order to obtain the enhanced feature $\widehat{\mathrm{x}_i}$, the weight distribution feature map ai of each channel is multiplied element-wise by the original feature map.

$$\widehat{\mathrm{x}_i} = \mathrm{x}_\mathrm{i} \cdot \sigma(\mathrm{a}_\mathrm{i}) \tag{5}$$

The spatial averaging function is considered to be a global representation of the channel scale. Each group of feature maps captures a specific semantics during the learning process. We use the global average feature to represent the learning characteristics of each group. Enhanced features can be obtained by using global features and local features for fusion. Our improved fire model just learns two parameters (λ, β) and adjusts the mask for feature enhancement.

3.3 Mask module

Our mask is inspired by image segmentation. Since CornerNet relies heavily on the generation of heatmaps, it is difficult to make predictions directly from the heatmap. We add a mask module to generate the split image directly. As described in Fig. 4, to achieve spatially enhanced features, the results of using the mask module are merged with the feature map. The mask constraint map increases the level of attention of the target object area. To simplify the process of feature extraction, we use a similar expectation-maximization algorithm to get the mask. This method uses the strategy of maximum likelihood estimation to get the mask. The internal features of the mask have the characteristics of low rank. This allows the mask to reduce the internal differences of the categories while maintaining the differences between the categories. The mask model rearranges the original features. The spatial distribution will change to facilitate heat map detection.

In order to ensure the simplicity of the original method, the mask module cannot be complicated. The Maximum Attention Mechanism (EMA) method abandoned the process of calculating the attention map on the full map [32]. The hidden variable is calculated iteratively using the expectation-maximization algorithm. And the attention mechanism is run on the hidden variable, which greatly reduces the complexity of the algorithm.

Inside the mask module, a 1 × 1 convolution kernel is used to squeeze the input feature channel to reduce the amount of calculation. Squeezing the input image channel can effectively use resources. Then upsample the feature map to the output

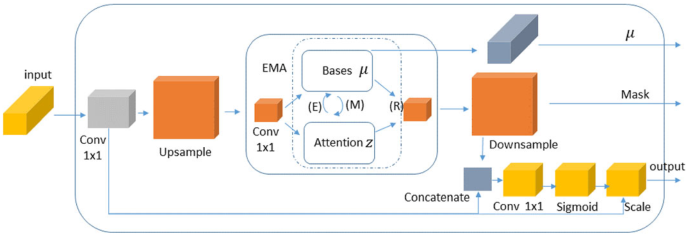

Fig. 4 Show the details of the mask module. Convolutional features are upsampled by multiple convolution kernels. Using the EMA module to estimate the mask, the mask module can get the segmentation result of the object. Aggregating the mask module can generate spatial constraints on the object and enhance local selection of features

size. The output size is half of the original image. Use EMA to calculate the feature map to get the output mask and bases parameter μ. The converged bases parameter μ and latent variables Z can be reconstructed output mask. Finally, the out mask and the original feature map are multiplied by the group elements to obtain the final feature. Out mask is the result of instance segmentation. Figure 5 is the result of object segmentation detection on some data.

The EMA module is composed of three parts: responsibility estimation (E), likelihood maximization (M), and data re-estimation (R). E and M is the E step and the M step of the EM algorithm. The size of the feature map X is C × H × W. To simplify the symbols, reshape X to N × C, where N = H × W. Then, feature map is $X \in R^{N\times C}$, the base initial value is $\mu \in R^{K\times C}$, and $Z \in R^{N\times K}$is a hidden variable. Ae estimates Z, and Am is used to update μ. Ae and Am alternately perform the T step to obtain an approximate estimate X of the feature. We set T to the default 3. Ar to get the features $\tilde{x}$ through μ and Z.

Step E calculates the posterior distribution of the hidden variable Z.

$$Z^{(t)} = soft\ \max\left(\lambda X\left(\mu^{(t-1)}\right)^{T}\right) \tag{6}$$

Step M updates μ by maximizing the likelihood function.

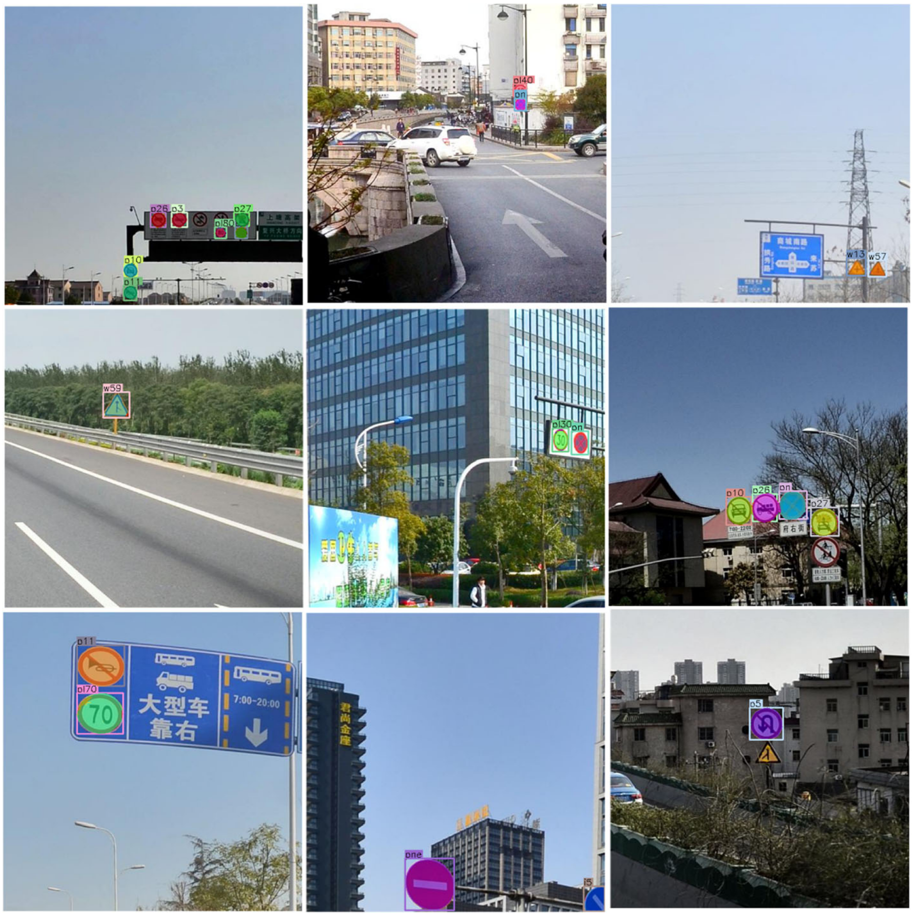

Fig. 5 The data tag is displayed, the left side is the mark for coordinate positioning, and the right side is the label for image segmentation

$$\mu^{(t)}{}_{k} = \frac{z^{(t)}{}_{nk} \cdot x_n}{\sum_{m=1}^{N} z_{mk}} \tag{7}$$

After E and M alternately perform the T step, μ and Z are used to reconstruct the feature map.

$$\tilde{x} = Z^T \cdot \mu^T \tag{8}$$

3.4 Training

L_{det} is the loss function for the regression of two diagonal corners of an object. L_{det} indicates that focal losses are used to constrain the position of the corner points. The category of the object is predicted by a non-standard Gaussian heatmaps. Locate a pair of coordinate points of the object through the top left and bottom right heatmaps.

L_{mask} is the loss function of the mask model. The mask module is an instance segmentation network, which is trained with a multi-class cross entropy loss function. The parameters of the mask module are updated using image segmentation.

L_{push} and L_{pull} are used to constrain the correlation between the corners of the same object and different objects. L_{pull} is used to constrain the distance between a pair of corner points of the same object. The distance between the corner points of the same object is the smallest. L_{push} is used to constrain the corner points of different objects, and the corner points of different objects are maximized.

L_{off} is used to compensate and correct the predicted and true value deviations, constrained by Smooth L1 Loss. Due to the large amount of down-sampling used in the convolutional network, the final coordinate regression and the coordinates in the original image are offset. Here we again predict an offset value to adjust the position of the angle. Here we use the smooth L1 Loss function as a penalty function.

The loss function for joint training for all tasks is as follows. α, β, λ, γ denote the weight of each task, the set value is 0.1, 0.1, 1,1.

$$\mathrm{L} = L_{\mathrm{det}} + \alpha L_{\mathrm{pull}} + \beta L_{\mathrm{push}} + \lambda L_{\mathrm{off}} + \gamma L_{\mathrm{mask}} \tag{9}$$

4 Experimental results and discussions

4.1 Datasets and implementation details

In this section, the experimental process and materials are detailed. We validated our method using the Tsinghua-Tencent 100K dataset [33]. The data set contains more than 10,000 images, of which there are 6000 training images and 3000 verification images. Three scales are set for the object scale, small, medium and large. The size of each object is [0–30], [30–96], and [96–400].

We analyzed the number of labeled objects in multiple small areas and the number of objects in each category. Detailed statistics are described in Fig. 6. The statistics of the number of individual categories are shown in the upper part of the figure. There are a total of 182 tag categories in the data set. Only the categories with more than 100 labels are shown in the figure. The lower part of the figure is the statistics of the size of the label object. Objects with size < 50

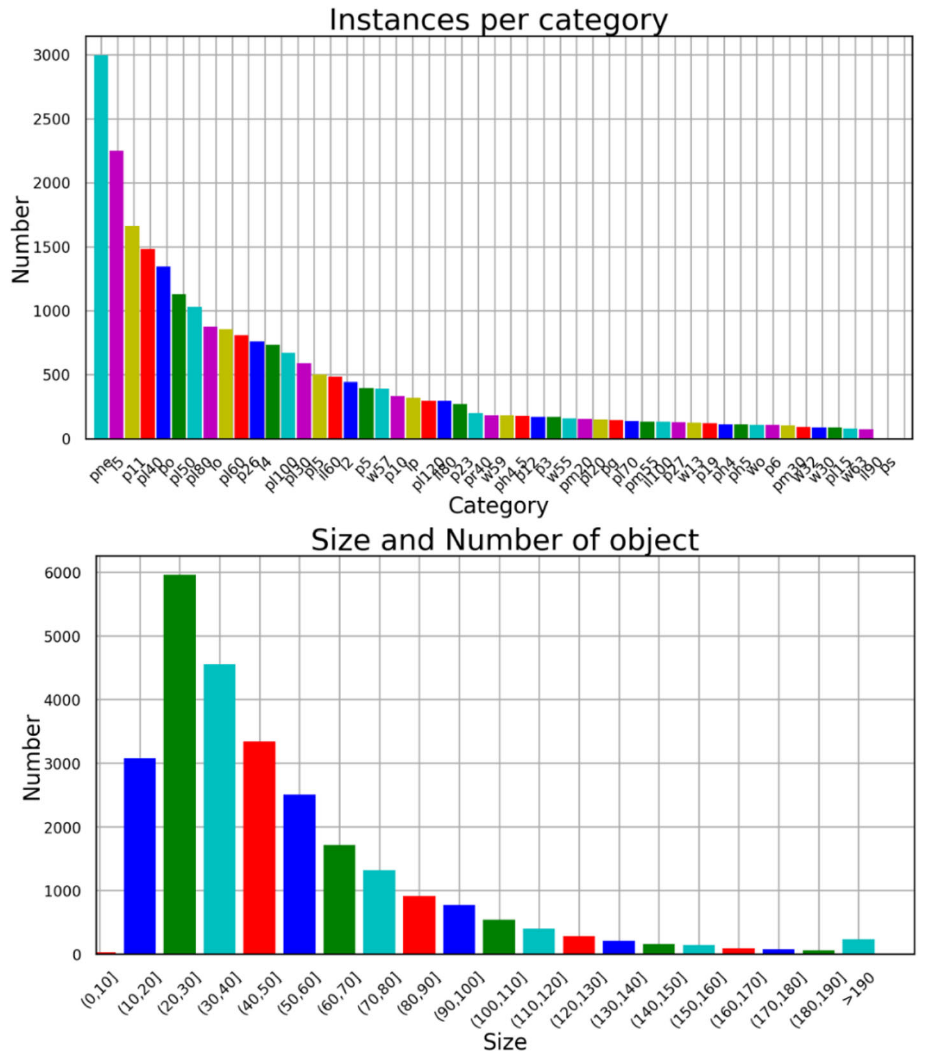

Fig. 6 Object size and category statistics

were found to account for 64.3% of all marked objects, while size ≥ 90 was only 11.3%. It can be clearly seen from the figure that small objects account for a large proportion in the data set. This makes it very challenging to detect this data set. This is very consistent with the actual car driving image. To reduce the intensity of training, we used the first 42 categories, which accounted for 89.5% of all tag categories.

Our method is implemented using Pytorch. The entire network parameters are randomly initialized. The size of the network input image is 512 × 512. The size of the heatmap output is 128 × 128. The output size of the mask is 256 × 256.We set both α and β to 0.1 and γ to 1. Training Mask-CornerNet using batch size = 13, learning_rate = 0.00025, the learning rate per 180,000 iteration is multiplied by 0.1. Use adam optimizes as the optimizer for the network. All experiments were verified on a NVIDIA GTX1080Ti GPU with 32GB memory workstation. In the model evaluation phase, select the top 100 top left corner and the top 100 bottom left corner in the heat map.

4.2 Results

In order to verify our method from multiple aspects, we compared the methods commonly used today, such as traffic sign detection methods based on real environment [33], using pyramidal convolutional networks to detect traffic signs [5], Faster-rcnn, yolov3 [12], CornerNet [25] , CornerNet-Squeeze and CornerNet-Saccade [14]. At the same time, in order to verify the effectiveness of each module from multiple sizes. Set multiple single optimized models based on the baseline method. CornerNet-fusion merges multiple features. CornerNet-fire configures the fire module of this article. CornerNet-mask contains the mask module. Mask-CornerNet is the method proposed in this paper. All reference object detection methods use the same data enhancement method. We study these object methods from three object scales: large, medium and small. Experimental results verify that the method in this paper has great advantages on multiple object scales.

The accuracy-recall curves of our method on multiple scales are shown in Fig. 7. Our Mask-CornerNet method performs well on small size and comprehensive size objects. The proposed method has an accuracy rate of 90.11% for small sizes and a recall rate of 88.42%. The accuracy of baseline in small-sized objects is 90.74%, and the recall rate is 86.33%. Pyramid convolutional networks do not have much advantage on small-sized objects. The accuracy and recall rate are 90.23 and 85.25%, respectively. Yolov3 detects multiple feature maps in small size with an accuracy rate and recall rate of 91.51 and 82.22%.

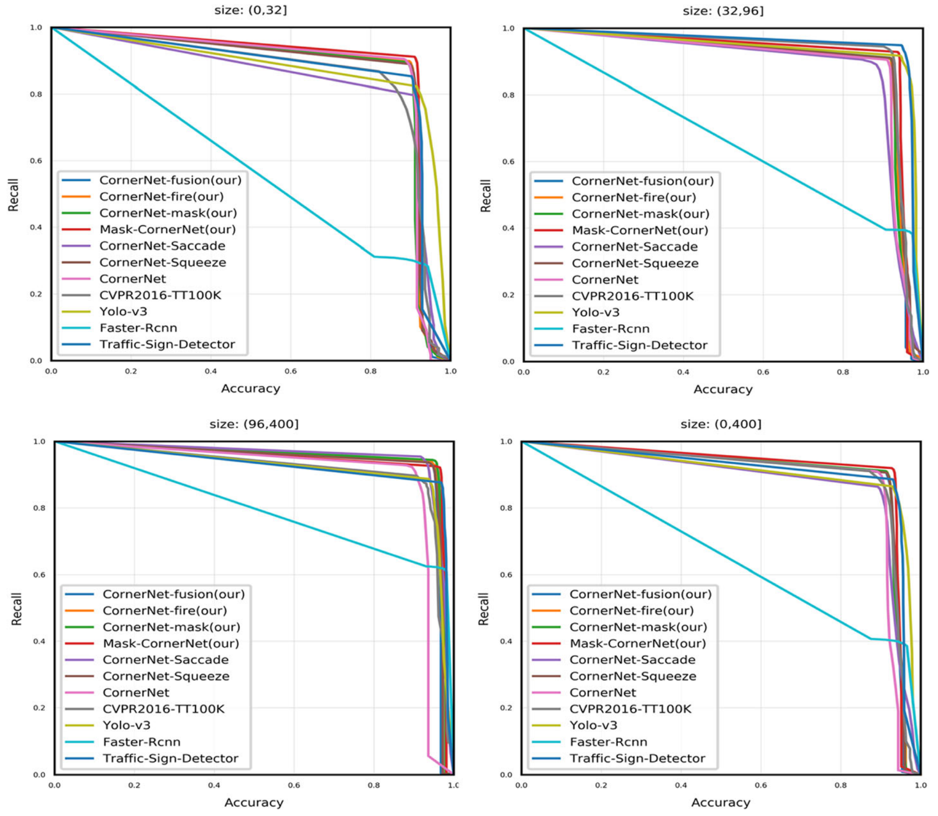

Fig. 7 The performance of the proposed method is on the three scales of large, medium and small

From the perspective of full size, the overall size accuracy and recall rate of our method are 91.63 and 89.83%. CornerNet has an accuracy of 90.29% and a recall of 89.22%. Conernet-Saccade has an accuracy of 90.33% at all scales and a recall of 84.14%. Our method improves the accuracy of comprehensive size detection by increasing the detection accuracy of small sizes.

4.3 Ablation study

We compare experiments by adding a single module to explore how much improvement each module can bring to object detection. All experiments were performed under the same conditions, environment, and hyper-parameters. This article uses the recall rate(R), accuracy rate (A), and mean average precision (mAP) as the evaluation criteria of the model. In this paper, mAP is the area enclosed by the Precision-recall curve and the *X*-axis. To take full advantage of contextual features, we use the output of a fusion of two hourglass networks. As can be seen from Table 1, the use of fusion features can significantly improve the network recall rate. The recall rate increased by 0.28%, while the accuracy rate decreased. After adding the fire model, it increased by 0.67% on the basis of the baseline. After adding the mask module, the accuracy of the model has increased by 0.71%. From the experimental results, all the added modules can improve the performance of object classification detection, thereby promoting the accuracy of the model. From mAP, our model's overall performance is optimized. The mAP of the proposed model is 90.01% vs. 89.84% of the baseline.

Our final model uses all the lifting methods, fuses contextual features, and optimizes the fire module to improve the segmentation network's ability to express features. Finally, on the baseline, we increased the recall rate by 1.04% with less cost.

In order to explore the specific influence of context on model feature expression, we perform object detection directly on the feature output of each hourglass network structure. The first layer is stack-1, the second layer is stack-2, and the fusion feature is stack-1 + stack-2. Table 2 shows our experimental results. The recall rate using the stack-1 model alone was 82.34%. The feature recall rate using fusion was 89.83%. It is able to increase by 7.49% on this basis. The tendency bias of recall rate and quasi-curvature was clearly balanced.

4.4 The effect of the number of proposals

Since Mask-Cornernet uses the top left and bottom right to detect objects separately, generally, the top 50–200 corner points with a score greater than the threshold are used. Use embedding vector similarity to match a pair of corners. We explored the

Table 1 Results of ablation experiments of various components

Baseline	Y	–	–	Y
+ fusion	–	Y	Y	Y
+ fire model	–	–	Y	Y
+ mask	–	–	–	Y
Accuracy	92.13	91.80	92.06	91. 63
Recall	88.79	89.07	89.46	89.83
mAP	89.84	89.72	89.91	90.01

Table 2 Comparison of the output results of the middle part features

Feature map	Stacks_1	Stacks_2	Stacks_1 + Stacks_2
Accuracy	92.95	91.72	91.63
Recall	82.34	89.76	89.83

influence of the number of corner points on the performance of object detection. Our experimental results are shown in Fig. 8. By adjusting the number of detection points in the experiment, it was found that the number of detection points did not affect the coordinate regression. This eliminates the impact of detection points on our algorithm.

4.5 Accuracy and efficiency

Our proposed method Mask-CornerNet is compared with ConerNet, YOLOv3, and RetinaNet in terms of time consumption and accuracy. Figure 9 shows the results of our experiment. Explore the performance of all methods on 6 different images of 128, 256, 448, 512, and 640. Our model size and schedule relationship are given in Table 3. Our fire-model can directly reduce the size of the entire CornerNet-squeeze by 112 Mb, while the accuracy is not reduced too much. After adding the Mask module, our model size has also been reduced to only 112.8 Mb. Our anchor-free object detection maintains its advantages in small-size object detection models. In this paper, the fire module and the mask module use a small number of parameters to optimize the baseline model. The increase in inference time is mainly caused by channel fusion and instance segmentation.

Figure 9 shows a comparison between our model and other models in terms of GPU consumption time. All experiments were performed in a consent environment. Due to the influence of the data set, the accuracy of all models on each input image size

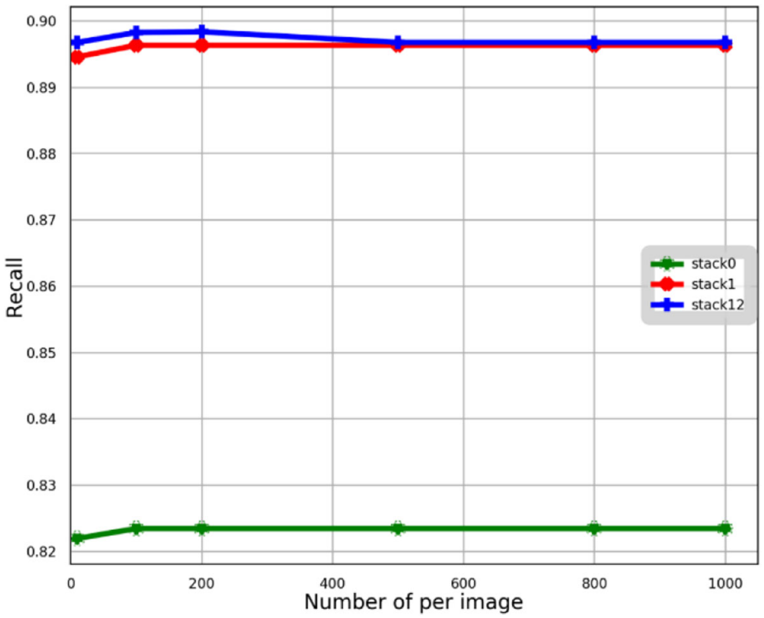

Fig. 8 Diagram the effect of the recommended number on the recall

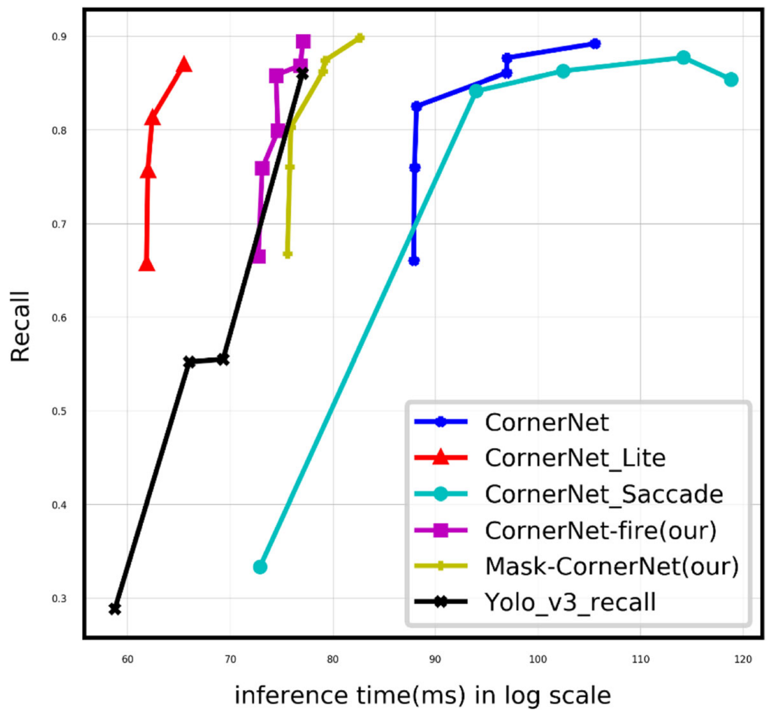

Fig. 9 Diagram the effect of the recommended number on the recall

fluctuates without reference value. So we just compare the time consumed by the GPU with the recall rate.

It takes 47 ms for YOLO-v3 to get a recall of 86.44%. CornerNet-squeeze takes 32 ms to the highest recall rate of 88.79%. After adding the fire module, the size of the model is significantly reduced. It takes only 41 ms for the recall rate to reach 85.82%, and 47.2 ms for 89.45%. Mask-CornerNet will take 52 ms to reach 87.21%, and it will take 62 ms to reach a recall rate of 89.82%. At the same time, Mask-ConerNet and ConerNet-fire are the same size. The CornerNet-Saccade and CornerNet models do not perform well on our dataset. CornerNet-Saccade takes 167 ms to obtain a recall rate of 86.27%, and it takes 301 ms to reach a recall rate of 87.70%. It takes 127 ms for CornerNet to reach a recall of 86.09%. And it takes 195 ms to reach a recall rate of 89.20%.

5 Conclusion

This paper proposes an anchor-free object detection method with mask attention mechanism. In order to make full use of the feature maps of convolutional networks, this paper fuses feature maps of multiple scales for object detection. The fire module proposed in this paper is beneficial to the model being transplanted to mobile devices.

Table 3 Time performance comparison result

	Time	Accuracy	Recall	Size
YOLO-v3	47 ms	92.90	86.44	247 Mb
CornerNet-squeeze	35 ms	92.13	88.79	128 Mb
CornerNet-fire (our)	41 ms	92.06	89.46	112 Mb
Mask-cornerNet (our)	62 ms	91.63	89.83	112.8 Mb

In this paper, the fire module can semantically group convolutional network channels to improve the function distribution of the convolution kernel. To directly use the convolutional network to generate the corner detection heat map is not ideal, this paper proposes to use the mask mechanism to guide the corner detection network. The instance segmentation results are fed back into the convolutional network to increase the diversity of convolutional network feature maps and enhance the network's ability to express. Our method was evaluated on the Tsinghua-Tencent 100 K dataset. And compared with the commonly used object detection method yolov3, and the latest CornerNet, CornerNet-squeeze method. Experimental results show that our method performs well on the data set.

The detection speed and model size of our model have room for further improvement. Next, we will design a more concise feature extraction network optimization model scale and detection accuracy. The mask module generates an instance segmentation with time consumption. We will further study the instance segmentation module with better time performance. This article mainly uses a traffic sign dataset containing a large number of small objects, and then we expand to the detection of other natural objects. This article also provides a feasible idea for improving the detection performance of traffic sign objects.

Abbreviations

R-CNN: Region with Convolutional Neural Network; YOLO: You only look once; SSD: Single Shot MultiBox Detector; FPN: Feature Pyramid Networks; R-FCN: Region-based Fully Convolutional Networks; U-Net: Convolutional Networks for Biomedical Image Segmentation; V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation; SGE: Spatial Group-wise Enhance

Acknowledgements

This research was financially supported by Shanghai Science and Technology Innovation Action Plan (No. 19DZ1207305) and Shanghai Key Laboratory of Advanced Manufacturing Environment. The authors express sincere appreciation to the anonymous referees for their helpful comments to improve the quality of the paper.

Authors' contributions

Beibei Fan led the project, provided topics, conducted reviews, and provided suggestions. He Yang did experiments and wrote this paper. Lingling Guo improves the paper. The author(s) read and approved the final manuscript.

Authors' information

1. Beibei Fan. She is now a professor at Shanghai University. Her research focuses on data mining, path planning, machine learning, and artificial intelligence.
2. He Yang. He is now a student of School of Mechatronic Engineering and Automation in Shanghai University. His research is mainly on data mining, machine learning, and image processing.
3. Lingling Guo. She is now a student of School of Mechatronic Engineering and Automation in Shanghai University. Her research is mainly on path planning and machine learning.

Funding

Not applicable.

Availability of data and materials

Tsinghua-Tencent 100 K dataset [10] is available at http://cg.cs.tsinghua.edu.cn/traffic-sign/.

Competing interests

The authors declare that they have no competing interests.

Received: 24 February 2020 Accepted: 6 July 2020
Published online: 14 July 2020

References

1. Z. Zou, Z. Shi, Y. Guo, J. Ye, in *arXiv e-prints*. Object detection in 20 years: a survey (2019) arXiv:1905.05055
2. W. Liu, I. Hasan, S. Liao, in *arXiv e-prints*. Center and scale prediction: a box-free approach for pedestrian and face detection (2019) arXiv:1904.02948
3. B. Yang, M. Tang, S. Chen, G. Wang, Y. Tan, B. Li, A vehicle tracking algorithm combining detector and tracker. EURASIP J. Image Video Process. **2020**, 17 (2020). https://doi.org/10.1186/s13640-020-00505-7
4. A. Sarwar, Z. Mehmood, T. Saba, K.A. Qazi, A. Adnan, H. Jamal, A novel method for content-based image retrieval to improve the effectiveness of the bag-of-words model using a support vector machine. J. Inf. Sci. **45**, 117–135 (2019)

5. Z. Liang, J. Shao, D. Zhang, L. Gao, Traffic sign detection and recognition based on pyramidal convolutional networks. Neural Comput. Appl. **32**, 6533–6543 (2020).
6. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. You only look once: unified, real-time object detection (2016), pp. 779–788. https://doi.org/10.1109/Cvpr.2016.91
7. S.Q. Ren, K.M. He, R. Girshick, J. Sun, in *Advances in neural information processing systems*. Faster R-CNN: towards real-time object detection with region proposal networks, vol 28 (2015)
8. K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, in *Proceedings of the IEEE international conference on computer vision*. Centernet: keypoint triplets for object detection, pp. 6569–6578 (2019)
9. H. Law, J. Deng, in *The European Conference on Computer Vision (ECCV)*. CornerNet: detecting objects as paired keypoints (2018)
10. L. Huang, Y. Yang, Y. Deng, Y. Yu, in *arXiv preprint*. DenseBox: unifying landmark localization with end to end object detection (2015) arXiv:1509.04874
11. J. Redmon, A. Farhadi, in *30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017)*. YOLO9000: better, faster, stronger (2017), pp. 6517–6525. https://doi.org/10.1109/Cvpr.2017.690
12. J. Redmon, A. Farhadi, in *arXiv preprint*. Yolov3: an incremental improvement (2018) arXiv:1804.02767
13. A. Newell, K.U. Yang, J. Deng, Stacked hourglass networks for human pose estimation. Lect. Notes Comput. Sci. **9912**, 483–499 (2016). https://doi.org/10.1007/978-3-319-46484-8_29
14. H. Law, Y. Teng, O. Russakovsky, J. Deng, in *arXiv preprint*. CornerNet-Lite: efficient keypoint based object detection (2019) arXiv:1904.08900
15. X. Li, X. Hu, J. Yang, in *CoRR*. Spatial group-wise enhance: improving semantic feature learning in convolutional networks (2019) abs/1905.09646
16. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, in *Proceedings of ECCV*. SSD: single shot multibox detector, pp. 21–37 (2016)
17. T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, S. Belongie, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Feature pyramid networks for object detection (2017)
18. T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, in *The IEEE International Conference on Computer Vision (ICCV)*. Focal loss for dense object detection (2017)
19. K. He, G. Gkioxari, P. Dollar, R. Girshick, in *The IEEE International Conference on Computer Vision (ICCV)*. Mask R-CNN (2017)
20. Dai, Y. Li, K. He, J. Sun, in Proceedings of advances in neural information processing systems. R-FCN: Object detection via region basedfully convolutional networks, pp. 379–387 (2016)
21. F. Yu, V. Koltun, in *arXiv e-prints*. Multi-scale context aggregation by dilated convolutions (2015) arXiv:1511.07122
22. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, in *arXiv e-prints*. MobileNets: efficient convolutional neural networks for mobile vision applications (2017) arXiv:1704.04861
23. J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, in *The IEEE International Conference on Computer Vision (ICCV)*. Deformable convolutional networks (2017)
24. Y. Li, Y. Chen, N. Wang, Z. Zhang, in *The IEEE International Conference on Computer Vision (ICCV)*. Scale-aware trident networks for object detection (2019)
25. H. Law, J. Deng, in *Proceedings of the European Conference on Computer Vision (ECCV)*. Cornernet: detecting objects as paired keypoints, pp. 734–750 (2018)
26. Z. Tian, C. Shen, H. Chen, T. He, in *Proceedings of Int. Conf. Computer Vision (ICCV)*. FCOS: fully convolutional one-stage object detection (2020)
27. X. Zhou, J. Zhuo, P. Krahenbuhl, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Bottom-up object detection by grouping extreme and center points (2019)
28. J. Long, E. Shelhamer, T. Darrell, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Fully convolutional networks for semantic segmentation, pp. 3431–3440 (2016)
29. O. Ronneberger, P. Fischer, T. Brox, in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*. U-Net: convolutional networks for biomedical image segmentation (Springer International Publishing, Cham, 2015), pp. 234–241
30. F. Milletari, N. Navab, S. Ahmadi, in *2016 fourth international conference on 3D Vision (3DV)*. V-Net: fully convolutional neural networks for volumetric medical image segmentation (2016), pp. 565–571
31. X. Li, X. Hu, J. Yang, in *arXiv preprint*. Spatial group-wise enhance: improving semantic feature learning in convolutional networks (2019) arXiv:1905.09646
32. X. Li, Z. Zhong, J. Wu, Y. Yang, Z. Lin, H. Liu, *Expectation-maximization attention networks for semantic segmentation* (2019)
33. Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, S. Hu, in *Proceedings of the IEEE conference on computer vision and pattern recognition*. Traffic-sign detection and classification in the wild, pp. 2110–2118 (2016)

Publisher's Note