



Yolov3-Pruning(transfer): real-time object detection algorithm based on transfer learning

Xiaoning Li¹ · Zhengzhong Wang¹ · Shichao Geng² · Lin Wang^{1,3} · Huaxiang Zhang^{1,3} · Li Liu^{1,3} · Donghua Li¹

Received: 11 March 2022 / Accepted: 7 May 2022 / Published online: 14 June 2022
© The Author(s) 2022

Abstract

In recent years, object detection algorithms have achieved great success in the field of machine vision. To pursue the detection accuracy of the model, the scale of the network is constantly increasing, which leads to the continuous increase in computational cost and a large requirement for memory. The larger network scale allows their execution to take a longer time, facing the balance between the detection accuracy and the speed of execution. Therefore, the developed algorithm is not suitable for real-time applications. To improve the detection performance of small targets, we propose a new method, the real-time object detection algorithm based on transfer learning. Based on the baseline Yolov3 model, pruning is done to reduce the scale of the model, and then migration learning is used to ensure the detection accuracy of the model. The object detection method using transfer learning achieves a good balance between detection accuracy and inference speed and is more conducive to the real-time processing of images. Through the evaluation of the dataset voc2007 + 2012, the experimental results show that the parameters of the Yolov3-Pruning(transfer): model are reduced by 3X compared with the baseline Yolov3 model, and the detection accuracy is improved, realizes real-time processing, and improves the detection accuracy.

Keywords Object detection · Transfer learning · Pruning · Detection accuracy · Inference speed · Real-time processing

1 Introduction

Under the wave of deep learning sweeping the world, deep neural networks [1] have made significant progress, solving the work from image classification to reinforcement learning, and the field of computer vision has received unprecedented attention. Object detection [2], as a challenging problem in this field, has also become one of the research topics for most researchers. Compared with deep learning programs such as image classification, object detection tasks are more complicated. Especially for images in different scenes, it is not only necessary to accurately locate the position of the object but also to determine the category to which the object belongs. In a complex scene [3], there may be more than one or two objects. The detection of multiple objects in the same image will be more complicated to solve. Therefore, object detection algorithms are bound to face problems such as a large amount of computation and inference delay [4] and cannot better realize real-time processing.

Many object detection algorithms are also constantly pursuing the balance between model accuracy and inference speed, such as Faster R-CNN [5], Yolo-tiny [6], Yolov3 [7], and a series of object detection algorithms. R-CNN [8] takes

✉ Lin Wang
wanglin_jn@hotmail.com

Xiaoning Li
lxn_luck@126.com

Zhengzhong Wang
2020317114@stu.sdu.edu.cn

Shichao Geng
gengsc@hotmail.com

Huaxiang Zhang
huaxzhang@hotmail.com

Li Liu
liuli_790209@163.com

Donghua Li
croseman@126.com

¹ School of Information Science and Engineering, Shandong Normal University, Jinan 250014, Shandong, China

² School of Journalism and Communication, Shandong Normal University, Jinan 250014, Shandong, China

³ Institute of Data Science and Technology, Shandong Normal University, Jinan 250014, Shandong, China

the object detection problem as the classification problem, extracts and classifies the features through the CNN model, and then recognizes the specific content through the RNN. Faster R-CNN [5], as the name suggests aims at making a breakthrough in latency, sharing the convolutional layer through the region proposal [9], then reducing a lot of calculations. The core of Yolo [10] is to treat the task of object detection as a regression problem. It inputs the image at one end, passes through the network framework, and obtains the desired boundary box coordinates and different category probabilities at the output layer, and it is an end-to-end model. Yolov3 uses the residual structure [11] on the network, taking the first 52 layers of convolution of darknet53 as the backbone network, which improves the model detection accuracy, but the scale of the model is also increased.

Large-scale target detection algorithms are more inclined to detect detailed information, while small-scale target detection algorithms are more inclined to semantic information. Mask-Refined R-CNN(MR R-CNN) [12] considered this problem and used mask optimization. It improves the accuracy of image segmentation obviously by combining the global and detailed information element maps. Chu et al. [13] proposed a target detection algorithm based on multi-layer convolution feature fusion (MCFF) and online hard case mining (OHEM). This method performs better in detecting small objects and occluded objects. In the process of detection, it optimizes the regional network through MCFF and then generates candidate regions. And OHEM algorithm is used to train the detector, which improves the training efficiency and accelerates the convergence speed.

Although many researchers are committed to reducing the delay of the program under the premise of ensuring the performance of the program, to achieve real-time processing. This research has not been done very well, especially in the field of object detection. Due to the diversity of objects and the complexity of the scene, reducing the scale of the model has received more and more attention in object detection. Traditional compression methods use techniques such as low-rank decomposition, pruning, and quantization to reduce model size. Although these methods reduce the model size to a certain extent and speed up the inference speed, they also reduce the accuracy of object detection. Therefore, it is difficult for object detection to achieve a balance between the accuracy and inference speed of the model. This is a problem that needs to be solved urgently and is also the research focus of our work.

To improve the inference speed of object detection algorithms and realize real-time processing of images, this paper proposes a fast and accurate object detection method based on transfer learning [14]. This method uses the teacher–student network [15] to train a more refined network model and use it in the object detection program. The student network (Yolov3-Pruning) learns the teacher network (baseline:

Yolov3) and has a good generalization ability. It can not only reduce the network model [16] but also improve the inference speed and detection accuracy of object detection, which can be better deployed in devices to achieve real-time performance. We can consider using this real-time object detection method in practical application scenarios, such as industrial bottle defect detection [17], to improve the speed of product defect detection in industrial production. This is very necessary for the quality monitoring process in the manufacturing industry.

The main contributions can be summarized as follows:

- The object detection algorithm is compressed by the model pruning algorithm.

We propose the Yolov3-Pruning object detection algorithm. It tests the pruning part of the model, comprehensively considers the time delay, parameters amount, and detection accuracy of the 13×13 , 26×26 and 52×52 feature layers, and selects the optimal result after pruning. Compared with the traditional Yolov3, it removes the 13×13 size feature layer, the convolutional layer of the backbone network [18] is reduced by nine layers, and the model parameters are reduced from 235.37 to 76.13 MB, which means the number of model parameters [19] is reduced by 3X.

- Establish the transfer learning of the teacher–student network.

To further improve the detection accuracy and the overall performance, we use migration learning. We take the baseline (Yolov3) as the teacher network, and take our Yolov3-Pruning as the student network, then use the parameters in the trained teacher network for self-training. Through transfer learning, the loss of detection accuracy of simple models is avoided. Compared with the unused Yolov3-Pruning(used), the Yolov3-Pruning(transfer) that uses the transfer is 20.29% higher in MAP-50; and 16.08% higher in MAP-75.

- Combining pruned models with transfer learning improves the real-time processing of images.

Yolov3-Pruning(transfer) under the voc2007 test set, the image processing speed reaches 43fps, which is 1.26X higher than the baseline Yolov3. It can be seen that our object detection algorithm is more conducive to the real-time processing of images.

The rest of the paper is organized as follows: Sect. 2 introduces related work on object detection algorithms and transfers learning. Section 3 describes our object detection algorithm in detail, including the model pruning algorithm for the baseline Yolov3 and the algorithm implementation

using the transfer learning design. Section 4 presents performance analysis with the baseline Yolov3 and experimental comparison results with other algorithms in the object detection area, while the conclusions of this paper are given in Sect. 5.

2 Related work

Nowadays, object detection in the field of computer vision has developed rapidly, and its algorithm structure is also constantly maturing. The object detection method can be introduced from two aspects, one is one-stage [20], and the other is two-stage [21]. One-stage only needs to classify and regress data blocks without generating candidate regions. It directly generates the position coordinates and category probabilities of the object. The prediction result can be obtained through one detection. Typical algorithms include SSD and Yolo. And two-stage refers to the classification and regression of data through two parts, first generating the candidate area and then classifying the candidate area, which will take more time. The typical algorithm is R-CNN.

The Yolo object detection algorithm in one-stage treats the image detection task as a single regression [22] problem. It takes the image as the input of the network model and obtains the bounding box coordinates and class probability in the output layer after a series of convolution operations. There are many different versions of the Yolo algorithm, and each time it is continuously improved, such as Yolov2, Yolov3, etc. Redmon et al. [23] proposed the Yolo framework in 2016. It is a regression problem with object detection boxes as spatially separated bounding boxes and predicted probabilities of classes. Yolov2 [24] is Yolo9000 proposed by Redmon et al. after Yolo, which can detect more than 9000 classifications. The system adopts the multi-scale training method, which can better balance the detection accuracy and inference speed of the model. Subsequently, Redmon et al. proposed improvements to Yolo object detection in 2018. The structure of the backbone network is adjusted, darknet53 is used, and multi-scale feature fusion is adopted.

Another typical one-stage algorithm, SSD, is a method for object detection based on a single deep neural network proposed by Liu et al. [25] in the field of object detection. SSD is detected in an end-to-end manner, retaining the design ideas of bounding box coordinates and category probability in Yolo detection and using multi-scale feature mapping [26]. And use VGG16 as the backbone network for feature extraction, object location [27], and classification recognition. This method uses the aspect ratio and feature size of each feature output position to calculate the confidence level for the object in each image and adjust the prediction frame appropriately to match the object in the image.

For the typical two-stage algorithm, Girshick et al. [28] proposed a simple and scalable object detection algorithm—R-CNN. The algorithm uses a convolutional neural network CNN, a top-down design, which is helpful for image segmentation [29] and detection. Moreover, when the datasets are insufficient, they can be supervised and pre-trained [30], and fine-tuned to specific domains to improve performance. Later they proposed a fast convolutional network Faster R-CNN [5]. This algorithm uses the VGG16 network to classify the feature extraction of the image, which not only improves the MAP value [31] but also significantly improves the training inference speed.

Currently, the use of hyperspectral (HS) [32] techniques in image classification tasks has gradually attracted widespread attention. The use of graph convolutional neural networks combined with hyperspectral for image classification will benefit object detection. Image remote sensing classification [33] is of great help for some geological surveys and can improve target detection performance in complex scenes that require fine classification. To better classify the image, a transformer can be used to reconsider hyperspectral image classification [34], with more subtle spectral differences to determine the recognized image. Use skip connections across layers to fuse residuals between layers for better detection.

As the algorithm continues to improve, the depth and width of the models are also growing. Therefore, many researchers are also working to find a balance between model accuracy and inference speed [35]. To improve the real-time performance of the object detection algorithm for image processing, it is necessary to compress the model under the premise of ensuring detection accuracy. Gui et al. [36] analyzed model compression by using different perspectives and considered compressing the model while maintaining accuracy and without compromising the robustness of the model against adversarial attacks. They proposed an adversarial training model compression framework—ATMC. The architecture designs a unified constrained optimization formula, which includes methods such as pruning [37], decomposition [38], and quantification [39]. Build a balance between the scale, accuracy, and robustness of the model so that it can be better applied to different devices. Moreover, the multi-network fusion algorithm and transfer learning of green cucumber segmentation and recognition in complex natural environments proposed by Bai et al. [40] are also worthy of our reference.

Object detection frameworks have also been changing in recent years. ORSim [41] is an object detection framework in optical remote sensing images based on time-frequency channel features. Channel feature extraction, learning, and image pyramid matching and boosting are integrated with this framework, all of which improve the level of object detection. Wu et al. [42] proposed a framework for object

detection in geospatial. The framework is based on Fourier Rotation-Invariant Features (FRIFB), first produced in polar coordinates, and then further refined on the aggregated channel features for boosted features. Subsequently, the target detection and tracking survey based on UAV was proposed [43]. To apply target detection to small devices such as drones, we must solve the problem of model scale and reduce the storage and complex calculation of target detection models. This is our current work.

Inspired by the above research results, we took the object detection algorithm Yolov3 as the baseline, modified it, and got our object detection algorithm. Among them, we use the model compression algorithm to reduce the convolutional layer of the model, and at the same time, use transfer learning to improve the detection accuracy of the model after compression, and improve the performance of real-time image processing.

3 Real-time object detection algorithm based on transfer learning

The overall framework structure of our proposed object detection algorithm based on transfer learning is shown in Fig. 1. The structure consists of two parts, one is the teacher network Yolov3 with a larger model, more parameters, and better accuracy, and the other part is the student network Yolov3-Pruning with 13×13 feature layers pruned. In the overall process, Yolov3 first conducts training to obtain a teacher network with higher detection accuracy. Then, the trained weights and bias parameters are used in Yolov3-Pruning to train the student network. It can ensure detection accuracy while reducing the network parameters and reach a good balance between the detection accuracy and delay of the model, which is conducive to real-time image processing.

3.1 Yolov3-Pruning

Yolov3-Pruning has made some changes based on Yolov3. The pruned model is shown in Fig. 2. Yolov3-Pruning reduces the number of network parameters by pruning the layer structure of the network model [44] and reducing the scale of the model. Before deleting the layer structure of the network, we analyzed the performance of each layer structure, selected the layer structure that has the least impact on the detection accuracy of the model, and pruned it.

3.1.1 Model pruning algorithm

Due to the increasing scale of convolutional neural networks, the model requires a lot of calculations and memory usage, and even the reasoning time is too long to achieve real-time performance.

Therefore, to make the object detection algorithm better applied, we made improvements based on the baseline Yolov3.

To reduce the size of the model and improve the inference speed, we design a pruning method. For the model pruning part, several different tests were done on the baseline Yolov3, and various performances were analyzed. On the Yolov3 object detection algorithm, we analyzed the whole process from the feature value extraction of the backbone network to the prediction result. We calculated the sum of three values of delay t , parameter m , and accuracy c for 13×13 , 26×26 , and 52×52 feature layers. The calculation formula is as follows:

$$\text{Total}_i = t_i + m_i + c_i, i \in (1, 2, 3) \quad (1)$$

Algorithm 1 :Model pruning in Yolov3.

Input:

H_{input} :height of feature layer input
 W_{input} :width of feature layer input
 H_{output} :height of feature layer output
 W_{output} :width of feature layer output
 t_i :delay of feature layer
 m_i :parameter quantity of feature layer
 c_i :accuracy of feature layer
 $channels_i$:the layers of the teacher network

Output:

$Total_i$: total evaluation index of feature layer

1: **Procedure**

2: //Obtain network models with pruned 13×13 , 26×26 and 52×52 feature layers

3: Execute training on each pruned model and obtain the detection accuracy c_i of each model.

4: Execute the test against the baseline Yolov3

5: **if** $H_{input} == 13$ **and** $W_{input} == 13$ **then**

6: Initialize t_1, m_1

7: **else if** $H_{input} == 26$ **and** $W_{input} == 26$ **then**

8: Initialize t_2, m_2

9: **else if** $H_{input} == 52$ **and** $W_{input} == 52$ **then**

10: Initialize t_3, m_3

11: **end if**

12: **if** $H_{output} == 13$ **and** $W_{output} == 13$ **and** $channels_i == 75$ **then**

13: $Total_1 = t_1 + m_1 + a_1$

14: **else if** $H_{output} == 26$ **and** $W_{output} == 26$ **and** $channels_i == 75$ **then**

15: $Total_2 = t_2 + m_2 + a_2$

16: **else if** $H_{output} == 52$ **and** $W_{output} == 52$ **and** $channels_i == 75$ **then**

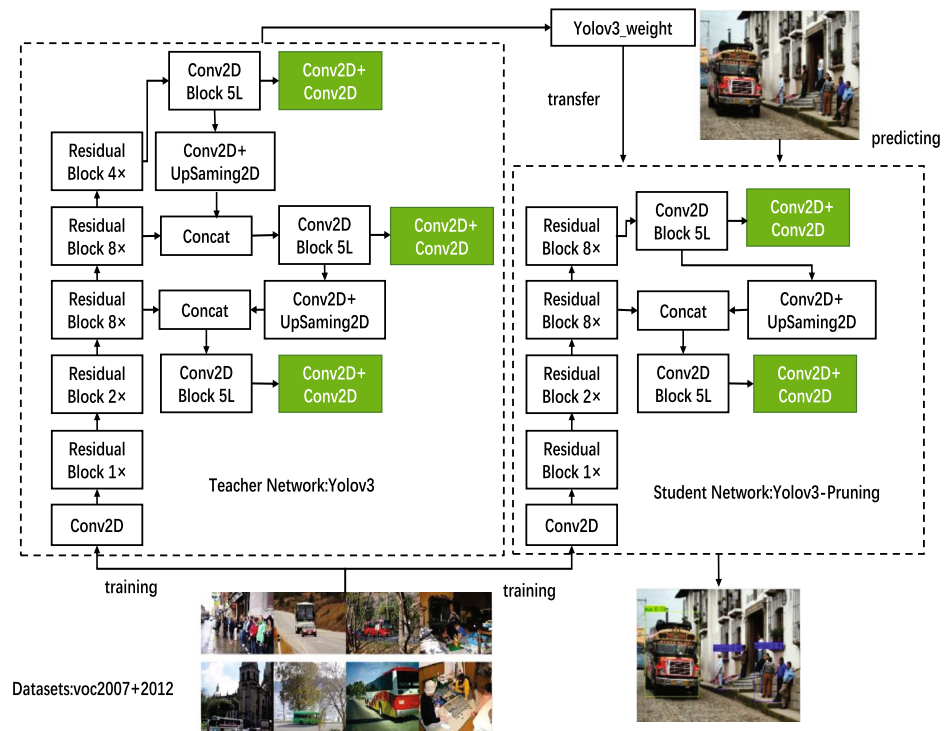
17: $Total_3 = t_3 + m_3 + a_3$

18: **end if**

19: **return** $\text{argmax}(Total_i)$

Algorithm 1 summarizes the model compression pruning process.

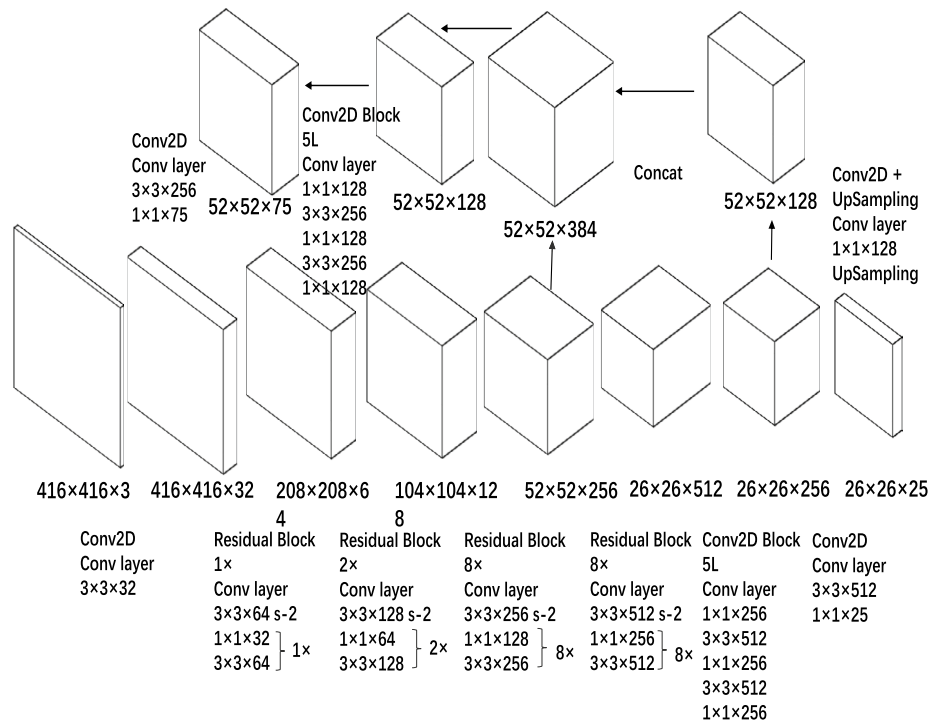
Fig. 1 The overall framework structure of the object detection algorithm is based on transfer learning. On the left is the baseline Yolov3, and on the right is the pruned Yolov3-Pruning. The small model learns the generalization ability of the large model and improves its accuracy



In the model compression process, the network models with pruned 13×13 , 26×26 , and 52×52 feature layers are obtained first. The pruned models are then trained, and the detection accuracy is obtained. Then, the baseline model is tested to obtain the delay in the execution of the three

feature layers and the number of parameters used. Finally, the maximum value of the sum of delay, parameter quantity, and detection accuracy is taken as the final result to decide which feature layer to select.

Fig. 2 Yolov3-Pruning network structure. The architecture prunes 13×13 feature layers and contains a total of 43 layers of convolution, reducing 9 layers of convolution computation compared to the baseline



Under the above compression pruning process, we choose to prune the 13×13 feature layer. The parameter amount of the model is reduced from 235.37 to 76.13 MB, which is 3X slower than the baseline model. Finally, we will take follow-up measures to further improve the detection accuracy of the model.

3.1.2 The distribution problem of the anchor box

The anchor box [45] is used to predict the object. It has a fixed width and height to constrain the ground-truth box and unify it to a fixed size. The expected box learns from the anchor box, retains the weights and biases, and finally converts it into the ground-truth box through translation or transformation. Our object detection algorithm prunes the feature layer based on the baseline Yolov3, and the anchor box should also be changed accordingly. Yolov3-Pruning and Yolov3 need to be compared under the same conditions, so we choose six anchor boxes in Yolov3 corresponding to 26×26 and 52×52 feature layers. Each group has three fixed anchor boxes with different widths and heights, and different anchor box sizes are related to the accuracy of object detection. Therefore, Yolov3-Pruning and Yolov3 use the same two sets of anchor boxes. Although the size of the anchor box is given fixedly, it is obtained by k -means [46] clustering, and the Euclidean distance [38] is used to calculate:

$$D = 1 - \text{IOU} \quad (2)$$

$$\text{IOU} = \frac{S_{\cap}}{S_{\cup}} \quad (3)$$

The initial anchor frame is random, where D is the distance between the anchor frame and the real frame, IOU is the intersection ratio of the real frame and the anchor frame, S_{\cap} is the coincident part of the anchor frame and the real frame, S_{\cup} is all parts of the anchor and ground-truth boxes.

Taking the 26×26 feature layer as an example, after passing our object detection algorithm, the shape of the output layer is (26, 26, 75). Where $75 = 3 \times (4 + 1 + 20)$, (26,26) is the width and height of the grid, 3 corresponds to three fixed anchor boxes, 4 corresponds to the center coordinates and width and height of the real box, 1 is whether the box contains objects, 20 is the 20 categories corresponding to the voc2007 + 2012 dataset. For the two sets of anchor boxes given, the anchor box most suitable for the real box will be selected as the final prediction result through the IOU (the maximum value of the intersection of the real box and the anchor box is used as the final anchor box). The anchor box distribution during the algorithm is shown in Fig. 3.

3.2 Design of transfer learning

After the Yolov3-Pruning object detection algorithm prunes a part, the detection accuracy of the model must be reduced. To improve the detection accuracy after pruning, we use the transfer learning method for reference. However, the design is different from traditional transfer learning.

The essence of transfer learning is to find the similarities to the original problem in the new problem to realize the transfer of knowledge. Between similar data or tasks, the previously learned model can be applied to different new fields. And we use the good performance and generalization ability of the original large-scale model to help the pruned model improve the detection accuracy.

Algorithm 2 :Transfer learning in teacher-student network.

Input:

D_{train} :the input training datasets of model
 $weight_T$:the weight of the teacher network
 $weight_S$:the weight of the student network
 $bias_T$:the bias of the teacher network
 $bias_S$:the bias of the student network
 len_T :convolution layers in teacher network
 len_S :convolution layers in student network
 $layer_T$:the layers of the teacher network
 $layer_S$:the layers of the student network

Output:

S_{pre} : pre-training parameters of the student network model

1: **Procedure**

2: //Obtain the network model parameters saved after the teacher network training

3: Execute training for D_{train} and obtain the $weight_T$ and $bias_T$ of each layer

4: **for** $i = 1; i < len_T; i ++$ **do**

5: **for** $j = 1; j < len_S; j ++$ **do**

6: **if** $layer_S == layer_T$ **then**

7: $bias_S = weight_T$

8: $bias_S = weight_T$

9: **end if**

10: **end for**

11: **end for**

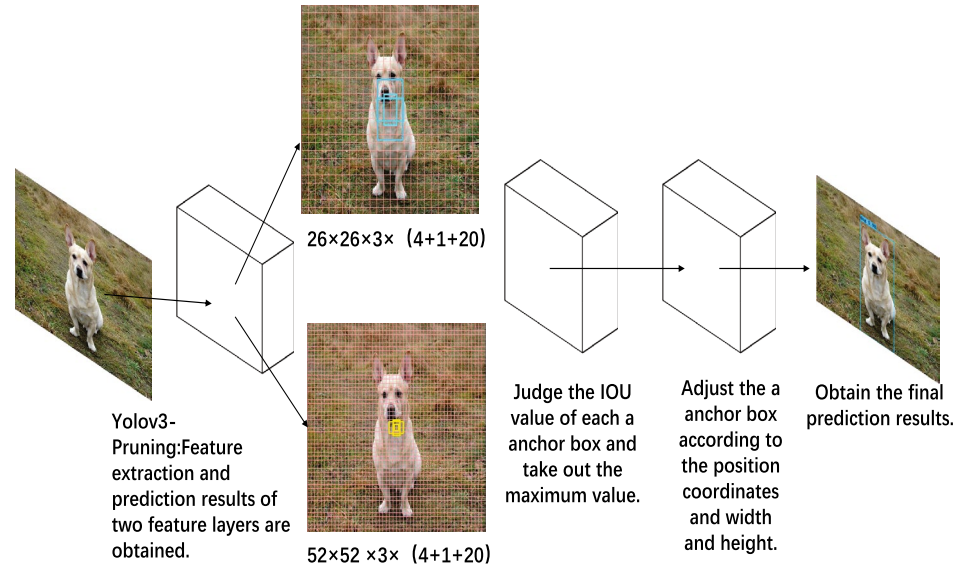
12: Execute the student network to get the predicted result S_{pre}

13: **return** S_{pre}

Algorithm 2 summarizes the migration process of our object detection algorithm.

During the migration process, we initialize the student network. The weights and biases of some layers in the teacher network are applied to the student network and

Fig. 3 Distribution of anchor boxes. As shown in the figure, the feature layers of 26×26 and 52×52 have three anchor boxes respectively. Through adjustment and calculation, the box with the best prediction effect is reserved



used as pre-trained model parameters, such as the following formulas:

$$\sum_{i=1}^N \text{weight}_S = \sum_{i=1}^N \text{weight}_T, N = \text{len}_s \tag{4}$$

$$\sum_{i=1}^N \text{bias}_S = \sum_{i=1}^N \text{bias}_T, N = \text{len}_s \tag{5}$$

4 Experiments

In the experiments, we use voc2007 and voc2012 training and validation dataset to train YOLOv3-Pruning. It is evaluated on the voc2007 test dataset and compared with other object detection algorithms.

4.1 Evaluation index

In the experiment, we made a detailed comparison of the object detection algorithms and also calculated the recall rate and accuracy rate. The specific definition is as follows:

$$\text{Recall} = \frac{TP}{TP + FP} \tag{6}$$

$$\text{Precision} = \frac{TP}{TP + FN} \tag{7}$$

Among them, TP is the number of positive samples predicted as positive samples, FP is the number of negative samples predicted as positive samples, and FN is the number

of positive samples that are incorrectly considered as negative samples.

4.2 YOLOv3-Pruning algorithm performance

We carry out ablation experiments to analyze YOLOv3-Pruning. A comprehensive analysis of the baseline was performed first, followed by an analysis of whether the pruned model used transfer learning or not. As can be seen from the data in Table 1, the number of parameters of the pruned YOLOv3 model has been significantly reduced, and it is 3X smaller than the baseline model, but the detection accuracy is also reduced a lot. When IOU= 0.5, it is reduced from 76.93% to 55.4%, with a decrease of 21.89%. When IOU= 0.75, it is reduced from 41.03% to 21.99%, with a decrease of 19.04%. To improve the detection accuracy of the pruned model, we use the transfer learning method. The detection accuracy of the YOLOv3-Pruning model based on transfer learning is as high as 75.33% (IOU= 0.5), which is higher than that of the YOLOv3-Pruning (unused) without transfer learning and is comparable to the baseline YOLOv3 detection accuracy.

From the visualization process of the three detection algorithms, baseline: YOLOv3 and YOLOv3-Pruning(transfer) is better than YOLOv3-Pruning(unused), as shown in Fig. 4. The prediction process graphs of the first two algorithms have clearer lines and more obvious outlines.

Figures 5 and 6 show the recall and precision curves of YOLOv3-Pruning and baseline YOLOv3 on the voc2007 test dataset, respectively. It can be seen from the figure that the recall rate of YOLOv3-Pruning (transfer) is between the baseline YOLOv3 and YOLOv3-Pruning (unused), and it is very close to YOLOv3 in multiple categories such as bottle, bus, and car; the degree is slightly higher than YOLOv3. As

for Yolov3-Pruning (unused), the recall rate is insufficient. Although the accuracy is similar, the fluctuation is too large, and there are uncontrollable factors.

Figure 7 shows the AP values of Yolov3-Pruning and the baseline Yolov3 in 20 categories (in the coordinate system, the recall rate is the horizontal axis, the precision is the vertical axis to get the PR curve, and the enclosed area is the AP). The MAP in the figure is the average value of various AP. It can be seen from the figure that the performance of the object detection algorithm that uses transfer learning is much higher than that of the object detection algorithm that does not use transfer learning, and it is closer to the baseline Yolov3. In categories such as bike, bus, chair, and tv-monitor, it is even slightly better than the baseline Yolov3. This shows that our model has improved its detection accuracy while learning Yolov3.

4.3 Image real-time processing comparison experiment

Figure 8 compares the real-time performance of three object detection algorithms. FLOPS stands for floating-point operations per second. It can be seen from the figure that baseline Yolov3 has the largest flops, indicating that it requires more data calculations and is more complex. Latency is the total time it takes for an image to be detected from the moment it is detected to the time it recognizes an object. The experimental data show that the latency of Yolov3-Pruning(transfer) and Yolov3-Pruning(unused) is the same, and both are lower than the baseline Yolov3, which indicates that our model is more conducive to real-time image processing. MAP represents the average detection accuracy of the model. FPS represents the number of image frames per second processed by the model. The more frames per second a model processes, the faster it infers, so the larger the value of FPS, the faster the inference speed. The faster the inference speed, the better the real-time processing power of the image. MAP and FPS are obtained under the voc2007 data. We performed tests on the entire dataset and obtained average results. It can be seen from the figure that the MAP of Yolov3-Pruning(transfer) is very close to the baseline Yolov3, which is much higher than that of Yolov3-Pruning(unused). And the FPS of

Yolov3-Pruning(transfer) is higher than the baseline Yolov3, indicating that the inference speed of Yolov3-Pruning(transfer) is faster, that is, its real-time performance for image processing is higher.

Figure 9 shows the inference time of each object detection algorithm. When the model executes, the shorter the inference time it takes, the faster it infers, and the better the real-time performance. We divide the voc2007 datasets into nine groups with 1107 images each. In each group, detection is performed on 1107 images, and the total inference time is obtained and averaged. It can be seen from the figure that Yolov3-Pruning(transfer) and Yolov3-Pruning(unused) take significantly less time than the baseline Yolov3, indicating that our object detection algorithm has a faster inference speed.

4.4 The comparative experiment of different object detection algorithms

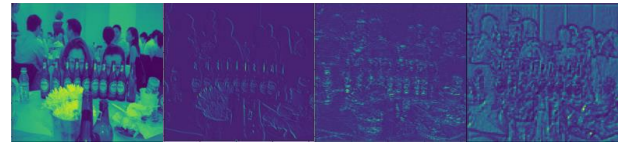
Table 2 shows the comparison between Yolov3-Pruning(transfer) and other algorithms (including ZLDN-L, WSOD², SDCN+FRCNN, Faster R-CNN). From the data in the table, we can see that the MAP of Yolov3-Pruning(transfer) is 2.2% higher than that of Faster R-CNN and 2.9% higher than that of SDCN+FRCNN, indicating that our model has a better target detection ability. It can be seen from the table that Yolov3-Pruning (transfer) can still maintain a high detection level after model pruning.

From Fig. 10, it can be seen that the average precision AP value of each classification in different target detection algorithms. We compared with ZLDN-L, WSOD², SDCN+FRCNN, and Faster R-CNN respectively. We can see that Yolov3-Pruning(transfer) is overall better than ZLDN-L, WSOD², SDCN+FRCNN, and Faster R-CNN in 20 categories. Faster R-CNN performs a little better in the cat category. In the bus category, the APs of the four algorithms are not much different, and the detection accuracy tends to be consistent. In the dog category, the three target detection algorithms of WSOD², SDCN+FRCNN, and Faster R-CNN are better than Yolov3-Pruning(transfer). However, in multiple categories, such as bike, boat, bottle, bus, car, chair, and person, Yolov3-Pruning(transfer) significantly outperforms other algorithms.

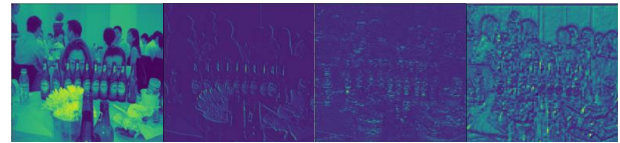
Table 1 The baseline Yolov3 and the pruned Yolov3-Pruning based on transfer learning are compared

Method	Parameters	MAP-50	MAP-75
Baseline: Yolov3	235.37 MB	76.93%	41.03%
Yolov3-Pruning(unused)	76.13 MB	55.04% (−21.89)	21.99% (−19.04)
Yolov3-Pruning(transfer)	76.13 MB	75.33% (−1.6)	38.07% (−2.96)

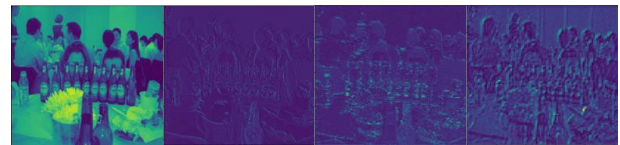
Fig. 4 Visualization of the prediction process for baseline Yolov3 and Yolov3-Pruning(transfer), Yolov3-Pruning(unused). Inputting the same image, the three methods respectively convert the image to a scale of 416×416 , and then make corresponding predictions



(a) baseline:Yolov3



(b)Yolov3 - Pruning(transfer)



(c)Yolov3 - Pruning(unused)

Fig. 5 Recall rate curve

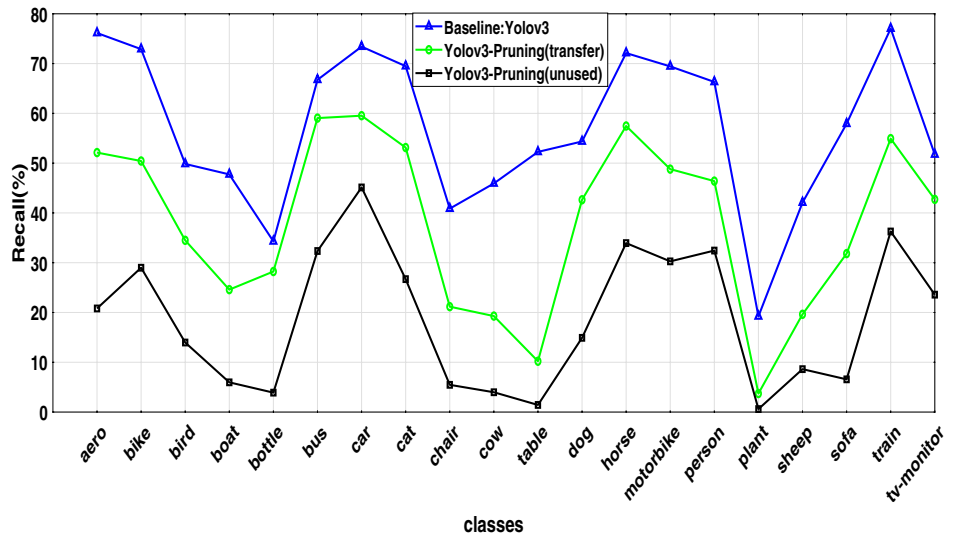
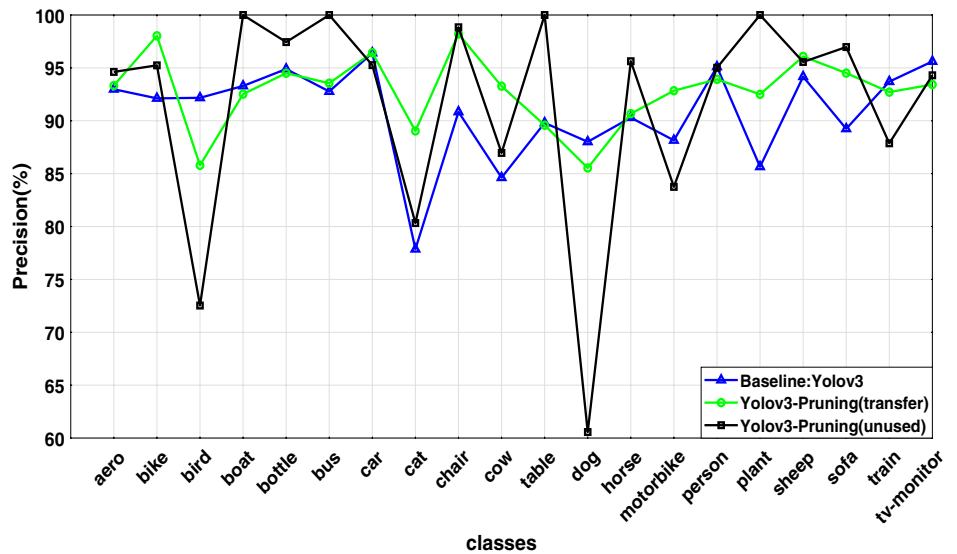


Fig. 6 Precision rate curve



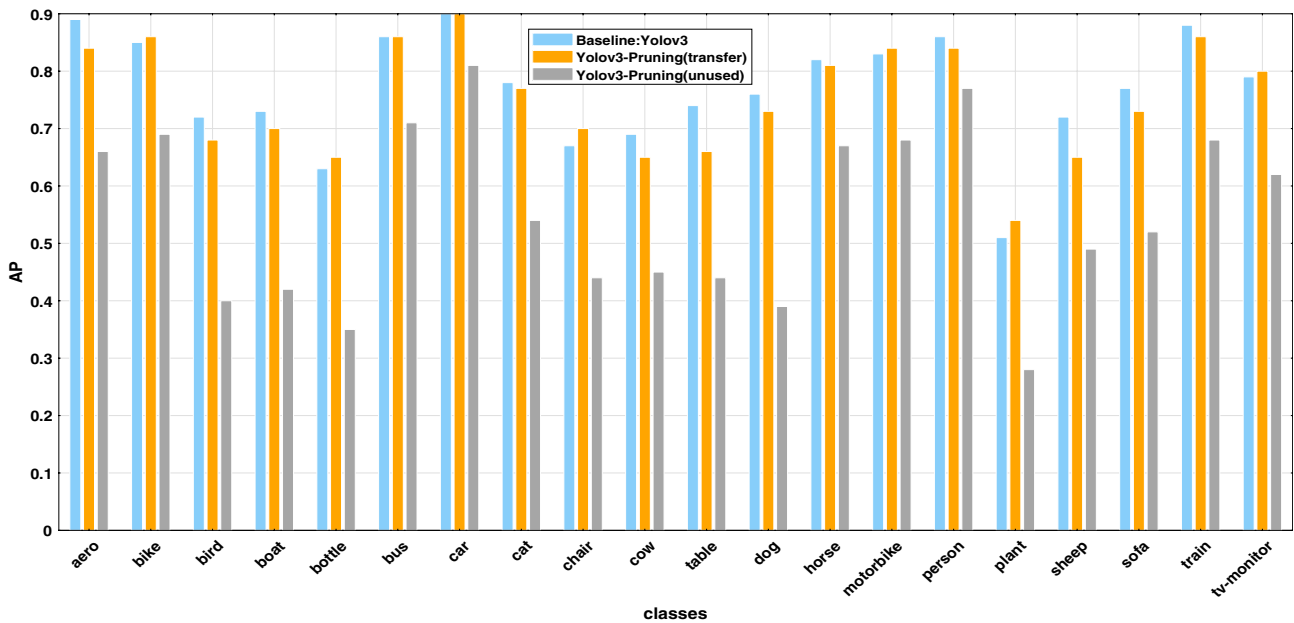


Fig. 7 Performance comparison of baseline Yolov3 and Yolov3-Pruning(transfer), Yolov3-Pruning(unused)

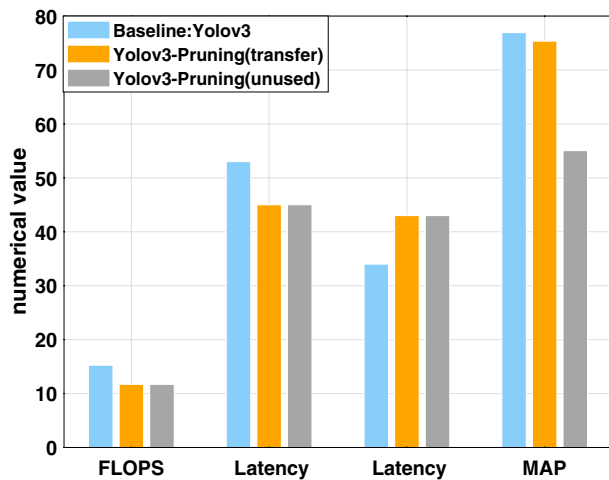


Fig. 8 Comprehensive comparison of algorithms in real-time image processing, including flops, latency, fps, and map. To be biased towards drawing, we set the unit of flops to be MIB/100 and the unit of latency to be ms

Table 2 The performance comparison of object detection algorithms under the voc2007+2012 dataset

Method	Train datasets	Test datasets	MAP
ZLDN-L [47]	voc2007+2012	voc2007	61.2
WSOD ² [48]	voc2007+2012	voc2007	69.5
SDCN+FRCNN [49]	voc2007+2012	voc2007	72.5
Faster R-CNN [50]	voc2007+2012	voc2007	73.2
Yolov3-Pruning(transfer)	voc2007+2012	voc2007	75.4

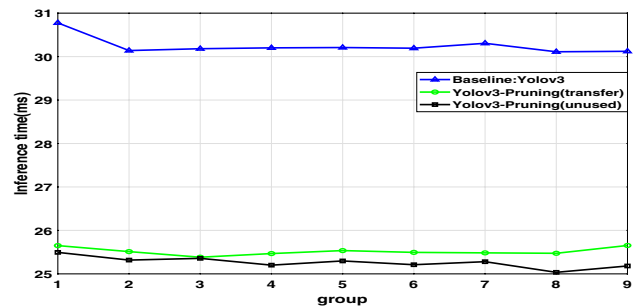


Fig. 9 Comparison of real-time processing performance of baseline Yolov3, Yolov3-Pruning(transfer), and Yolov3-Pruning(unused) algorithms (the abscissa in the figure is nine groups, and the ordinate is the average inference time)

4.5 Predicted results

After modification by Yolov3-Pruning, images are randomly selected for detection on voc2007. The predicted results are shown in Fig. 11. From the three images in the first row, it can be seen that the prediction effect of Yolov3-Pruning (transfer) based on transfer learning is between the baseline Yolov3 and Yolov3-Pruning (unused) without transfer learning. Yolov3-Pruning did not detect the table, indicating

that the model with the 13×13 feature layer trimmed is not suitable for detecting large objects. In the three images in the second row, you can see that Yolov3-Pruning (transfer) has detected a small object (car). In the three images in the

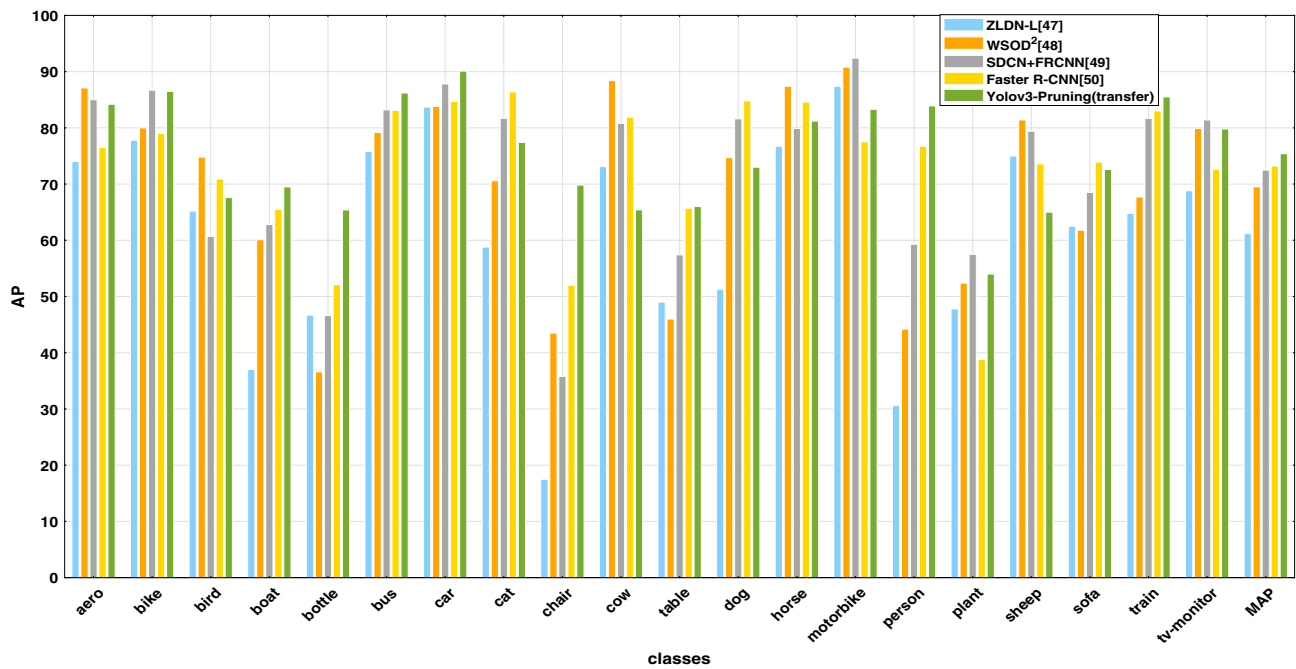
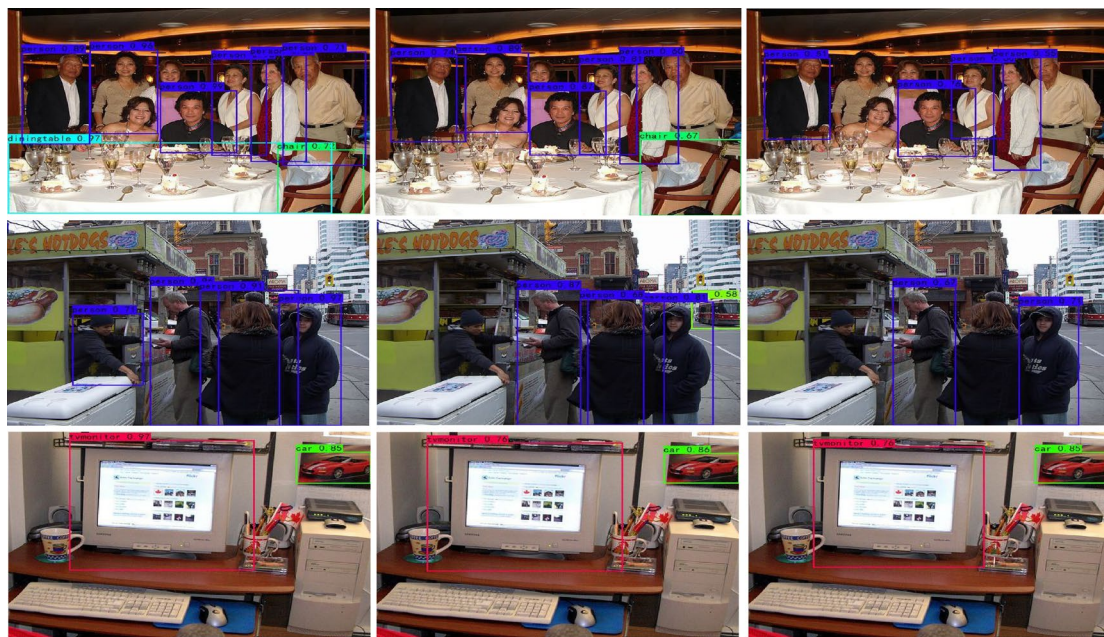


Fig. 10 AP values of different object detection algorithms in each category. During the experiment, we used 20 classifications and obtained detection results under the voc2007 dataset



(a)Baseline: Yolov3 (b)Yolov3 - Pruning (transfer) (c) Yolov3 - Pruning (unused)

Fig. 11 The prediction results of the selected images in the voc2007 dataset. We randomly selected three images in the dataset to compare the prediction results of each object detection algorithm. Looking at

the information in the figure, we can see that our model is more suitable for detecting small objects

third row, two kinds of objects are detected, but Yolov3-Pruning (transfer) detects small objects (car) on the wall with a slightly higher detection accuracy by one percentage

point. It can be seen from this that our object detection algorithm based on transfer learning performs well in detection accuracy and is conducive to the detection of small objects.

5 Conclusion

In this work, the object detection algorithm based on transfer learning is implemented. Yolov3-Pruning is a further improvement of the baseline Yolov3. The baseline Yolov3 structure is trimmed to reduce the scale of the model and significantly improve the image detection inference speed and help to realize real-time image processing. As the model size decreases, the detection accuracy also decreases. To overcome this problem, we use the transfer learning method to ensure the detection accuracy of the model. Compared with baseline Yolov3, the network structure of the YOLOv3-Pruning algorithm is simple, easy to set up, and the number of parameters is small, which helps the object detection algorithm to achieve real-time performance. Finally, the experiment shows that the object detection algorithm based on migration learning that we have achieved has achieved good performance in detection accuracy and inference speed and has also reached a good balance between the two, improving the performance of real-time image processing.

In the future, we will further study the algorithm to improve the performance of the model. We can choose several of the model compression algorithms and combine them. Improve the accuracy of the model from different aspects, and analyze the performance of the algorithm from different perspectives to achieve better real-time processing. Based on reducing the size of the model, we will make it applicable to practical scenarios, such as remote sensing image detection, and target detection and tracking of UAVs.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Chen, Z., Xu, T.-B., Du, C., Liu, C.-L., He, H.: Dynamical channel pruning by conditional accuracy change for deep neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**(2), 799–813 (2020)
- Wang, L., Tang, J., Liao, Q.: A study on radar target detection based on deep neural networks. *IEEE Sensors Lett.* **3**(3), 1–4 (2019)
- Javed, S., Mahmood, A., Al-Maadeed, S., Bouwmans, T., Jung, S.K.: Moving object detection in complex scene using spatiotemporal structured-sparse rpca. *IEEE Trans. Image Process.* **28**(2), 1007–1022 (2018)
- Millon, M., Galan, A., Courbin, F., Treu, T., Suyu, S., Ding, X., Birrer, S., Chen, G.-F., Shajib, A., Sluse, D., et al.: TDCOSMO-I. An exploration of systematic uncertainties in the inference of H0 from time-delay cosmography. *Astron. Astrophys.* **639**, 101 (2020)
- Lee, C., Kim, H.J., Oh, K.W.: Comparison of faster R-CNN models for object detection. In: 2016 16th International Conference on Control, Automation and Systems (ICCAS), pp. 107–110. IEEE (2016)
- Oltean, G., Florea, C., Orghidan, R., Oltean, V.: Towards real time vehicle counting using yolo-tiny and fast motion estimation. In: 2019 IEEE 25th International Symposium for Design and Technology in Electronic Packaging (SIITME), pp. 240–243. IEEE (2019)
- Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint [arXiv:1804.02767](https://arxiv.org/abs/1804.02767) (2018)
- Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)
- Wang, J., Chen, K., Yang, S., Loy, C.C., Lin, D.: Region proposal by guided anchoring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2965–2974 (2019)
- Chen, Q., Wang, Y., Yang, T., Zhang, X., Cheng, J., Sun, J.: You only look one-level feature. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13039–13048 (2021)
- Tai, Y., Yang, J., Liu, X.: Image super-resolution via deep recursive residual network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3147–3155 (2017)
- Zhang, Y., Chu, J., Leng, L., et al.: Mask-refined R-CNN: a network for refining object details in instance segmentation. *Sensors* **20**(4), 1010 (2020)
- Chu, J., Guo, Z., Leng, L.: Object detection based on multi-layer convolution feature fusion and online hard example mining. *IEEE Access* **6**, 19959–19967 (2018)
- Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., Liu, C.: A survey on deep transfer learning. In: International Conference on Artificial Neural Networks, pp. 270–279 (2018). Springer
- Fang, W., Xue, F., Ding, Y., Xiong, N., Leung, V.C.: EdgeKE: an on-demand deep learning IoT system for cognitive big data on industrial edge devices. *IEEE Trans. Ind. Inf.* **17**(9), 6144–6152 (2020)
- He, Y., Zhang, X., Sun, J.: Channel pruning for accelerating very deep neural networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1389–1397 (2017)
- Patel N, Mukherjee S, Ying L. Erel-net: A remedy for industrial bottle defect detection. In: International Conference on Smart Multimedia. Springer, Cham, pp. 448–456 (2018)
- Valueva, M.V., Nagornov, N., Lyakhov, P.A., Valuev, G.V., Chervyakov, N.I.: Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Math. Comput. Simul.* **177**, 232–243 (2020)
- Chen, X., Yu, K.: Hybridizing cuckoo search algorithm with biogeography based optimization for estimating photovoltaic model parameters. *Sol. Energy* **180**, 192–206 (2019)
- Tian, Z., Shen, C., Chen, H., He, T.: Fcos: fully convolutional onestage object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 9627–9636 (2019)
- Rajendran, K., Mahapatra, D., Venkatraman, A.V., Muthuswamy, S., Pugazhendhi, A.: Advancing anaerobic digestion through two-stage processes: current developments and future trends. *Renew. Sustain. Energy Rev.* **123**, 109746 (2020)

22. Shen, X.-J., Dong, Y., Gou, J.-P., Zhan, Y.-Z., Fan, J.: Least squares kernel ensemble regression in reproducing kernel hilbert space. *Neurocomputing* **311**, 235–244 (2018)
23. 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)
24. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263–7271 (2017)
25. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: single shot multibox detector. In: *European Conference on Computer Vision*, pp. 21–37 (2016). Springer
26. Li, X., Zhang, W., Ding, Q.: Deep learning-based remaining useful life estimation of bearings using multi-scale feature extraction. *Reliabil. Eng. Syst. Saf.* **182**, 208–218 (2019)
27. Huang, C.-Q., Yang, S.-M., Pan, Y., Lai, H.-J.: Object-location-aware hashing for multi-label image retrieval via automatic mask learning. *IEEE Trans. Image Process.* **27**(9), 4490–4502 (2018)
28. 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)
29. Minaee, S., Boykov, Y.Y., Porikli, F., Plaza, A.J., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell* (2021)
30. Chen, T., Frankle, J., Chang, S., Liu, S., Zhang, Y., Carbin, M., Wang, Z.: The lottery tickets hypothesis for supervised and self-supervised pretraining in computer vision models. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16306–16316 (2021)
31. Li, C., Yang, T., Zhu, S., Chen, C., Guan, S.: Density map guided object detection in aerial images. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 190–191 (2020)
32. Hong, D., Gao, L., Yao, J., et al.: Graph convolutional networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **59**(7), 5966–5978 (2020)
33. Hong, D., Gao, L., Yokoya, N., et al.: More diverse means better: multimodal deep learning meets remote-sensing imagery classification. *IEEE Trans. Geosci. Remote Sens.* **59**(5), 4340–4354 (2020)
34. Hong D, Han Z, Yao J, et al. SpectralFormer: rethinking hyperspectral image classification with transformers. *IEEE Trans. Geosci. Remote Sens.* (2021)
35. Ji, R., Cao, L., Wang, Y.: Joint depth and semantic inference from a single image via elastic conditional random field. *Pattern Recogn.* **59**, 268–281 (2016)
36. Gui, S., Wang, H.N., Yang, H., Yu, C., Wang, Z., Liu, J.: Model compression with adversarial robustness: unified optimization framework. *Adv. Neural. Inf. Process. Syst.* **32**, 1285–1296 (2019)
37. Luo, J.-H., Wu, J.: Autopruner: an end-to-end trainable filter pruning method for efficient deep model inference. *Pattern Recogn.* **107**, 107461 (2020)
38. Yang, H.-F., Chen, Y.-P.P.: Hybrid deep learning and empirical mode decomposition model for time series applications. *Expert Syst. Appl.* **120**, 128–138 (2019)
39. Xiao, H., Cinnella, P.: Quantification of model uncertainty in rans simulations: A review. *Prog. Aerosp. Sci.* **108**, 1–31 (2019)
40. Bai, Y., Guo, Y., Zhang, Q., et al.: Multi-network fusion algorithm with transfer learning for green cucumber segmentation and recognition under complex natural environment[J]. *Comput. Electron. Agric.* **194**, 106789 (2022)
41. Wu, X., Hong, D., Tian, J., et al.: ORSIm detector: a novel object detection framework in optical remote sensing imagery using spatial-frequency channel features[J]. *IEEE Trans. Geosci. Remote Sens.* **57**(7), 5146–5158 (2019)
42. Wu, X., Hong, D., Chanussot, J., et al.: Fourier-based rotation-invariant feature boosting: an efficient framework for geospatial object detection. *IEEE Geosci. Remote Sens. Lett.* **17**(2), 302–306 (2019)
43. Wu X, Li W, Hong D, et al. Deep learning for UAV-based object detection and tracking: a survey. *arXiv preprint arXiv:2110.12638* (2021)
44. Liu, Z., Sun, M., Zhou, T., Huang, G., Darrell, T.: Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270* (2018)
45. Zhong, Y., Wang, J., Peng, J., Zhang, L.: Anchor box optimization for object detection. In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1286–1294 (2020)
46. Yu, S.-S., Chu, S.-W., Wang, C.-M., Chan, Y.-K., Chang, T.-C.: Two improved k-means algorithms. *Appl. Soft Comput.* **68**, 747–755 (2018)
47. Zhang X, Feng J, Xiong H, et al.: Zigzag learning for weakly supervised object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4262–4270 (2018)
48. Zeng Z, Liu B, Fu J, et al. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8292–8300 (2019)
49. Li X, Kan M, Shan S, et al.: Weakly supervised object detection with segmentation collaboration. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9735–9744 (2019)
50. Wang, G., Guo, J., Chen, Y., et al.: A PSO and BFO-based learning strategy applied to faster R-CNN for object detection in autonomous driving. *IEEE Access* **7**, 18840–18859 (2019)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xiaoning Li is currently pursuing an M.S. degree in the School of Information Science and Engineering from Shandong Normal University. Her research interests include machine learning and program performance optimization.

Zhengzhong Wang is currently pursuing an M.S degree in the School of Information Science and Engineering from Shandong Normal University. His research interests include machine learning.

Shichao Geng graduated from Beihang University with a major in computer application technology in 2016 and entered the School of Journalism and Media of Shandong Normal University in September of the same year. He has published nine papers, applied for two invention patents, and participated in many vertical and horizontal projects such as the National Natural Science Foundation of China, the Doctoral Fund of the Ministry of Education, and the intelligent diagnosis of faults in CGNPC nuclear power plants. Research direction: deep learning (including intelligent diagnosis, art application direction), uncertainty artificial intelligence.

Lin Wang received her PH.D. degree from School of Computer Science and Engineering, Beihang University, China. From 2017, she was an assistant professor with Shandong normal university, and handled the China Postdoctoral Science Foundation and National Natural Science Foundation of China. She has worked in the areas of computer architecture and deep learning.

Huaxiang Zhang received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2004. He was an Associated Professor with the Department of Computer Science, Shandong Normal University, Jinan, China, from 2004 to 2005. He is currently a professor with the School of Information Science and Engineering, Shandong Normal University, together with the School of Information Science and Electrical Engineering, Shandong Jiaotong University. He has authored over 200 journal and conference papers and has been granted 31 invention patents, and is supported by the program of “Taishan Scholar”. His current research interests include machine learning, pattern recognition, evolutionary computation, and multimedia analysis.

Li Liu received the Ph.D. degree (2007) at Shandong University. Now she is a professor and master supervisor of School of Information Science and Engineering, Shandong Normal University. She has authored more than 40 academic papers. Her research interests include computer graphics, machine learning and pattern recognition.

Donghua Li is currently pursuing an M.S. degree in the School of Information Science and Engineering from Shandong Normal University. His research interests include performance optimization and cache partition.