A Multi-scale Bagging YOLO for Object Detection

Xiaoqiang Lu, Guojin Cao, Xinyu Liu, Zixiao Zhang, Yuting Yang
School of Artificial Intelligence, Xidian University
Xi'an, Shannxi Province, 710071, China
{luxiaoqiang5903, caoguojincn, 17788066762, zhangzixiao1999}@163.com, Ytyang_1@stu.xidian.edu.cn

Abstract

In this report, we introduce the technical details of our submission to the VIPriors object detection challenge. Our solution is based on frameworks of YOLOv4[1], YOLOv5, and their variants with EMA. Firstly, we adopt a sophisticated training strategy that could help us obtain 4 independent Validation sets. Secondly,we introduce an effective data augmentation method to address the lack of data problem, which contains mosaic[1], mix-up[18] and random color-jitter. Thirdly, we utilize a multi-model integration strategy to refinement the prediction box, which weighted boxes fusion (WBF)[16]. Furthermore, both multi-scale training[14] and multi-scale testing are included, which have significant detection performance for small targets. Experimental results demonstrate that our approach can significantly improve the AP @ 0.50:0.95 to 0.305 on the DelftBikes1[4] dataset.

1. Introduction

In order to save training data and reduce energy consumption, the 2nd Visual Inductive Priors for Data-Efficient Deep Learning Workshop was introduced as an ICCV 2021 workshop to promote data efficiency. As part of the workshop, five challenges were offered covering various popular research areas of computer vision such as image classification, object detection, instance segmentation, action recognition, re-identification. Each challenge uses a small fraction of the publicly available benchmark dataset with the strict rule that models must be trained from scratch and no external data is allowed.

Object Detection has made great process in recent years. Most of the current state-of-art detectors[15, 10, 12, 7, 11] are finetuned from huge amount of annotated data[13].In many practical applications cenarios,due to the limitations of various conditions,we cannot get a large number of samples for training.Therefore,it is necessary to study object detection algorithm based on small samples.

Existing object detection datasets such as MS-COCO[8], Imagenet-det[13], and Open Image[6] have no annotated object parts. Pascal-Parts[2] include part labels, yet lack information if a part is missing and where, as is required to evaluate visual verification. Thus, [4] shows how popular object detectors hallucinate objects in a visual part verification task and introduce the first visual part verification dataset: DelftBikes1, which has 10,000 bike photographs, with 22 densely annotated parts per image, where some parts may be missing. It focuses on obtaining high average precision (AP) on a DelftBikes1 object detection dataset.

To address this challenge, we focus on data augmentation in data preprocess and utilize the mosaic[1] and mixup methods to improve the diversity of object images. Then we respectively trained yolov4, yolov5, and multiple variants for them to detect. Finally, we adopt weighted boxes fusion (WBF)[16] method of the multiple-model ensemble on the test dataset. The network framework is shown in Figure 1. Experimental results demonstrate our approach can significantly improve the object detection performance and achieve a competitive result on the test set. The implementation details of the above are described in section 2 and section 3.

2. Methods

2.1. Data Augmentation

DelftBikes1[4] is a dataset of 10k images with 22 densely annotated parts specifically collected and labeled for visual verification. The dataset is randomly split into 8k for training and 2k for testing. To increase the number of samples to train, we have utilized mosaic[1] and mixup argumentation methods.

Mosaic[1] argumentation is proposed as part of the YOLOv4 pipeline, which works very similar to that of stitchers that four sub-images make up one image to be used as a sample, except that each sub-image is cropped by a random size selected from a range of scales. In this challenge, we incorporate one more trick that uses the supercategory information as prior knowledge.

The general data augmentation method is to transform

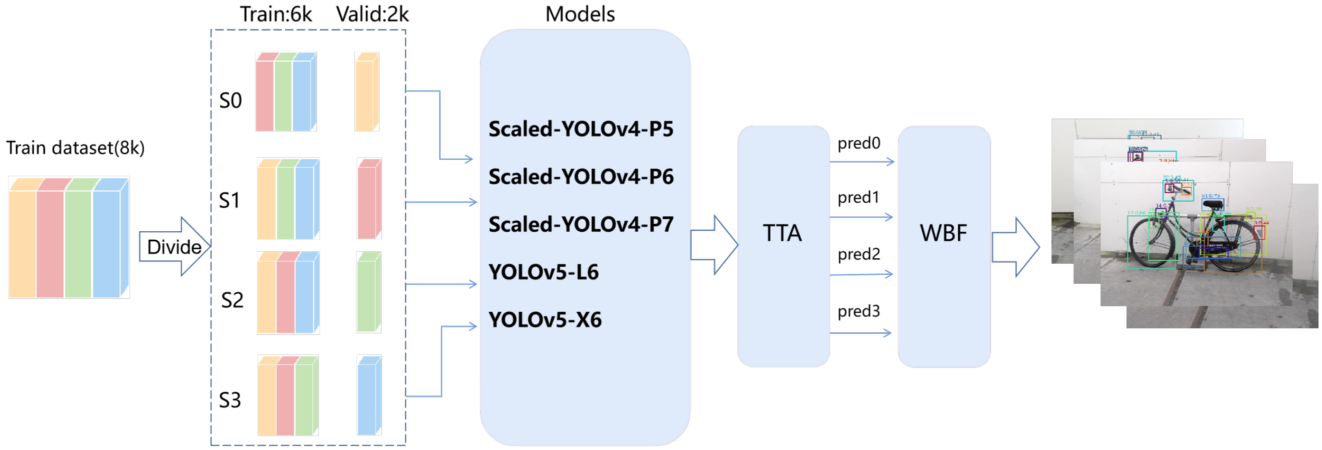

Figure 1. The overall architecture of our framework architecture.

the same category. A mix-up[18] uses modeling between different categories to achieve data augmentation. Two samples are randomly selected to improve the diversity of the training set from the training samples for simple random weighted summation. At the same time, the labels of the samples also correspond to the weighted summation, and then the prediction results and the weighted summation of the label after the loss are calculated, and the parameters are updated in the backpropagation.

2.2. Weighted Boxes Fusion

Many model fusion methods can significantly improve performance, such as voting, boosting, bagging, and stacking. In this report, we adopt a weighted boxes fusion (WBF) integration algorithm that improves detection performance by integrating predictions of different object detection models. The goal of WBF is to fuse the information of all prediction frames. It can correct a situation in which all models predict the frame to be inaccurate.

3. Experiments

3.1. Implement Detail

As shown in Figure 1, our solution is based on the framework of Scaled-YOLOv4[17], YOLOv5, and their several variants.

All experiments were conducted using the Scaled-YOLOv4[17], YOLOv5 toolbox developed by PyTorch[9]. And we run experiments on NVIDIA V100. Training configurations are summarized as follows. (1) According to the multi-scale training strategy, we scale the input images to different sizes, such as 1280, 1408, 1536, 1664, 1792. (2) For both mix-up[18] and random color jittering, the implementation probability is set to 0.5. (3) The mini-batch stochastic gradient descent (SGD) method[5] is applied to optimize parameters. The weight decays are set to 5×10^{-4}, and the momentums are set to 0.937. (5) Because of the train from Scratch, the maximum epoch for training is set to 100 and the batch size is 32.

3.2. Training Strategy

According to a previous paper, we know that multi-scale training[14] is a very effective method to improve the results, so we did not use experiments to verify this conclusion again, but directly used it.

The training set has a total of 8k images. For the small sample data set, we divide it into 4 equal parts in an orderly manner. There are 2k images in each part, which are respectively used as the verification set of sub-data S0, S1, S2, and S3. Therefore, four independent training data subsets are obtained. The above four training sets are used as input, and the weights are obtained through the detector. After that, the multi-scale test is performed, and the image size is respectively resized into 1280, 1408, 1536, 1664, 1792. And we will get values of pred0, pred1, pred2, pred3. Finally, the final prediction result is obtained after they pass through weighted boxes fusion (WBF)[16].

3.3. Evaluating visual verification

For visual verification, we want high recall of present parts and low recall of missing parts where detecting the same object multiple times does not matter. Besides, wrongly detected missing parts (false positives) cost more than not detected present parts (false negatives). Thus, our F_{vv} evaluation score is based on recall and inspired by the F_β score[3] so we can weight detection mistakes differently as

$$F_{vv} = \frac{\left(1+\beta^2\right) R^P \left(1-R^M\right)}{\beta^2 \left(1-R^M\right) + R^P} \tag{1}$$

Method	AP @ 0.50:0.95
YOLOv4-P5+muilti-scale	0.2954
YOLOv4-P5+muilti-scale+*	0.2989
YOLOv4-P6+muilti-scale	-
YOLOv4-P6+muilti-scale+*	-
YOLOv4-P7+muilti-scale	-
YOLOv4-P7+muilti-scale+*	-
YOLOv5-L6+muilti-scale	-
YOLOv5-L6+muilti-scale+*	0.2974
YOLOv5-X6+muilti-scale	-
YOLOv5-X6+muilti-scale+*	-
Ensemble 200 models	0.305

Table 1. Test phase submission results.

R^P is the present recall and R^M the missing recall calculated at a certain IoU threshold. The β parameter allows to weight the detection mistakes, where we set the β parameter to 0.1 so that detections of missing parts are 10x more costly than not detected present parts.

3.4. Results

Results are shown in Table 1. Comparing the two baselines of YOLOv4[1] and YOLOv5, scaled-yolov4 performs better. Among the models of the Scaled-YOLOv4 series, Scaled-YOLOv4-p5 has the best performance. We speculate that it is because the target size of the training data is moderate but uniform, and there are not too many detection targets of extremely small size.

4. Conclusion

This report details the key technologies used in the VIPriors object detection challenge. Our primary concern is data augmentation to extract more compelling features. The introduction of mosaic[1], mix-up[18], color-jitter techniques to expand the training set make the model more robust. Besides, a strong and effective model architecture is also very important. Our selection of Scaled-YOLOv4[17], YOLOv5, and their variants have shown a powerful ability to predict missing objections. Finally, we adapt to a weighted boxes fusion (WBF)[16] method of the multiple-model ensemble on the test dataset to modify the prediction box, and the AP @ 0.50:0.95 after model integration reaches 0.305.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection, 2020.

[2] Xianjie Chen, Roozbeh Mottaghi, Xiaobai Liu, Sanja Fidler, Raquel Urtasun, and Alan Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts, 2014.

[3] Nancy A. Chinchor. Muc-4 evaluation metrics. In *MUC*, 1992.

[4] Osman Semih Kayhan, Bart Vredebregt, and Jan C. Van Gemert. Hallucination in object detection – a study in visual part verification. 2021.

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

[6] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, and et al. The open images dataset v4. *International Journal of Computer Vision*, 128(7):1956–1981, Mar 2020.

[7] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[8] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[10] S. Qiao, L. C. Chen, and A. Yuille. Detectors: Detecting objects with recursive feature pyramid and switchable atrous convolution. *arXiv*, 2020.

[11] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[12] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Bernstein. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 1–42, 2014.

[14] Bharat Singh, Mahyar Najibi, and Larry S. Davis. Sniper: Efficient multi-scale training, 2018.

[15] R Solovyev, W. Wang, and T. Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. 2019.

[16] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. Weighted boxes fusion: Ensembling boxes from different object detection models. *Image and Vision Computing*, 107:104117, Mar 2021.

[17] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Scaled-yolov4: Scaling cross stage partial network, 2021.

[18] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018.