Research Towards Yolo-Series Algorithms: Comparison and Analysis of Object Detection Models for Real-Time UAV Applications

Research Towards Yolo-Series Algorithms: Comparison and Analysis of Object Detection Models for Real-Time UAV Applications

Shuo Wang[1,*]

[1]Department of Electrical and Computer Engineering, Indiana University Purdue University Indianapolis, Indianapolis, IN 46202, USA

[*]sw126@iu.edu

Abstract-General Unmanned Aerial Vehicles (UAVs) are widely used through the computer vision functions of onboard cameras and embedded systems. However, due to the limited memory and computing power of embedded devices on the UAV platform, it is a very challenging issue to analyze the real-time scene through the object detection method. To deal with these challenges, this paper compares the performance of different Yolo series models on the Pascal VOC dataset, using mAP and FPS as evaluation metrics, and applies the training results to the XTDrone UAV Simulation Platform for testing. We evaluate YOLOv3, YOLOv3-tiny, YOLOv3-SPP3, YOLOv4, and YOLOv4-tiny on the Pascal VOC benchmark dataset; the mAP of YOLOv4 is 87.48%, which is 14.2% higher than that of YOLOv3. FPS reaches 72, and the test time on the test set is 103.86s; the shortest test time on the validation set is Yolov4-tiny, but the mAP only reaches 50.06%, which is not as good as Yolov3-tiny. This paper compares the performance of five models in the Pascal VOC Dataset and simulates them on the XTDrone platform, and finally concludes that Yolov3-Tiny can meet the requirements of real-time, lightweight, and high precision.

1. Introduction and Related Works

With the rapid development of UAV and computer vision, UAV aerial images have been widely used in military and civil fields, such as reconnaissance and search, terrain exploration, air-ground coordination [1]. UAV platforms are required to be able to parse and react scene accordingly, of which the core part is parsing the scene. These scene resolution functions correspond to tasks in computer vision, namely image classification, object detection, and semantic segmentation [2,3]. Due to the complex background, large interference, fewer pixels occupied by the target, and the rapid movement of UAV in the shooting process, aerial object detection is more complex than ordinary images [4, 5]. At the same time, there are also difficulties in object detection algorithms on UAV platforms. But it is difficult to balance detection accuracy and real-time performance when extracting appropriate target features manually. It is difficult to get good results by traditional methods. A competitive way to solve these problems is the object detector based on deep learning technology.

With the increase of computing power and the emergence of large-scale labeled sample data sets, a deep neural network has been widely studied for its fast, scalable, and end-to-end learning framework. Especially, compared with the traditional methods, the convolutional neural network (CNN) [6] model has been significantly improved in image classification (such as ResNet [7] and DenseNet [8]), object detection (such as Faster R-CNN [9] and SSD [10]) and semantic segmentation (such as UNet [11] and Mask R-CNN [12]). Since the CNN model was successfully introduced into the object detection task

(R-CNN) [13], this detection framework has attracted extensive research interest. In the past five years, there have been many new target detectors based on CNN. The detector based on deep learning can be divided into two types: two-stage and one-stage.

1.1 Two-stage detectors

The two-stage method refers to the method of generating some alternative boxes in a certain way, then classifying the contents in the alternative boxes and modifying the positions of the alternative boxes. Because it contains two steps, region proposal, and detection, it is called a two-stage method. R-CNN adopts a strategy based on regional recommendations [14], in which each proposal is scaled before using Convnet for classification. More accurate detectors such as Fast R-CNN [15] and Faster R-CNN advocate the use of features calculated from a single scale because of its good performance in balancing accuracy and processing time. But the processing time still can't meet the requirements of the embedded board. Also, it is difficult to use on UAV due to the complexity of a large amount of memory and network.

1.2 One-stage detectors

Considering the high efficiency of the one-stage object detection method, Liu et al. proposed SSD method [16], in which ConvNet was used to spread anchors of different scales to multiple layers and forced each layer to predict the target on a certain scale. Fu et al. proposed a deconvolution single time detector (DSSD), which combined Resolution-101 with SSD and expanded it through the deconvolution layer to introduce more large-scale background for object detection and improve detection accuracy. Among these one-stage detectors, YOLO series model [17-19] may be the fastest target detection algorithm with the most advanced detection accuracy. On a graphics processing unit (GPU) card with high performance computing power, YOLO series models' real-time performance reported in literature is evaluated. YOLO uses a forward convolutional network to predict the class and position of objects at speeds up to 45 FPS. Then, YOLOv2 [18] is proposed to improve YOLO in several aspects, such as using the high-resolution layer, adding batch normalization on each convolutional layer, and using the convolutional layer with anchor box instead of the fully connected layer to predict the boundary box. With the development of the basic network, YOLOv3 [19], which uses Darknet-53 to replace the backbone network and uses multi-scale features to detect the target was proposed. Yolo series model (Joseph Redmon et al.) is the most popular deep target detector in practical application, because the detection accuracy and speed are very balanced. However, these detectors' reasoning still needs high-performance computing and large runtime memory to maintain good detection performance, which brings high computational cost and power consumption platform to UAV embedded devices. Therefore, how to choose the appropriate model to deploy on a UAV platform has become an urgent problem [20].

In this paper, we propose to compare the performance of YOLOv3 [19], YOLOv3-tiny [19], YOLOv3-SPP3 [19], YOLOv4 [21] and YOLOv4-tiny [21] models, and analyze the models that are most suitable for deployment on UAV platform. We evaluate the above models on Pascal VOC dataset [22]; and YOLOv3-tiny achieves a convincing result: in the case of ensuring accuracy, it also takes very little time.

2. Methodology

2.1 YOLOv3

YOLOv3 is the third version of the YOLO (You Only Look Once) series of object detection algorithms. Compared with the previous algorithms, the accuracy has been significantly improved. For each of the input images, YOLOv3 is going to predict three different sized 3D tensors, which will correspond to three different scales which can detect objects of different sizes. For the scale, the original input image is divided into 13x13 Grid cells, each of which is equal to 1x1x255, the rectangular pieces of the voxel that are drawn across the 3D tensor. The 255 figure comes from (3x(4+1+80)), whose numbers represent the Bounding box's coordinates, the object Score, and the confidence of each corresponding class. Second, if the bounding box corresponding to a certain ground truth in the training set happens to fall

within an image input of a grid cell (the red Grid cell in the figure), the Grid cell is responsible for the prediction of the object's bounding box, so the Grid cell's object Score is assigned 1, and the rest of the Grid cell is 0. In addition, each Grid cell is assigned three prior boxes of different sizes. During the learning process, the Grid cell will gradually learn how to select the prior box size and fine-tune the Prior box (offset/coordinate). There is a rule, which only selects the prior box with the highest conformity with the Ground Truth Bounding box IOU.

YOLOv3 introduces residual module on the basis of darknet-19, and further deepens the network. The improved network has 53 convolution layers, named darknet-53. The network structure is as follow:

Table 1. Residual module darknet-53 network structure

	Type	Filters	Size	Output
	Convolutional	32	3 x 3	256 x 256
	Convolutional	64	3 x 3 / 2	128 x 128
1 x	Convolutional	32	1 x 1	
	Convolutional	64	3 x 3	
	Residual			128 x 128
	Convolutional	128	3 x 3 / 2	64 x 64
2 x	Convolutional	64	1 x 1	
	Convolutional	128	3 x 3	
	Residual			64 x 64
	Convolutional	256	3 x 3 / 2	32 x 32
8x	Convolutional	128	1 x 1	
	Convolutional	256	3 x 3	
	Residual			32 x 32
	Convolutional	512	3 x 3 / 2	16 x 16
8x	Convolutional	256	1 x 1	
	Convolutional	512	3 x 3	
	Residual			16 x 16
	Convolutional	1024	3 x 3 / 2	8 x 8
4x	Convolutional	512	1 x 1	
	Convolutional	1024	3 x 3	
	Residual			8 x 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

2.2 YOLOv3-SPP3 (Compared with YOLOv3)

YOLOv3-SPP3 introduce the SPP module [23]. The module is as follow:

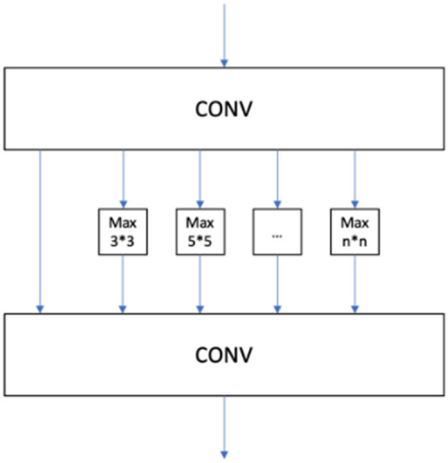

Figure 1. SPP Module

In general CNN network structure, the classification of the last layer is usually composed of all connections, and the connection has a characteristic, which is the characteristic number is fixed, this leads to the image in the input network, the size must be fixed, but in reality, the image size is varied, if can't meet the network's input, images will not be able to forward in network operations, so in order to get a fixed size of the image, must be cut or images such as tensile deformation, so it is likely to lead to image distortion, which affects the final accuracy, We want the network to be able to keep the input size of the original image with maximum accuracy. The additional feature channels introduced by SPP modules as well as extra FLOPs can be reduced and refined by channel pruning afterwards. In our experiments with VOC dataset, we add a SPP module in YOLOv3 between the 5th and 6th convolutional layers in front of each detection header to formulate YOLOv3-SPP3.

2.3 YOLOv3-tiny (Compared with YOLOv3)

Yolo3-tiny is a simplified version of YOLOv3. The main difference is that the backbone network uses 7 convolution layers and Max networks to extract features (similar to darknet19), while the grafted network uses 13*13 and 26*26 resolution detection networks. The main advantages of YOLOv3-tiny are that the network is simple, the calculation is small, and it can run on the mobile terminal or the device side [24][25]. The disadvantage is that the accuracy is relatively low (both candidate frame and classification accuracy are relatively low).

2.4 YOLOv4 (Compared with YOLOv3)

CSPDarknet53 serves as the backbone network, SPP serves as the additional module of Neck, PANet [26] serves as the feature fusion module of Neck, and YOLOv3 serves as Head. CSPDarknet53 [27] added CSPNet (Cross Stage Partial Network) on each large residual block of Darknet53 and integrated it into the feature map through gradient change. The feature map was divided into convolution operation and the other for combination with the last convolution result. CSP can effectively improve CNN's learning ability and reduce calculation. Path Aggregation Network makes full use of feature fusion. In YOLOv4, the fusion method is changed from addition to multiplication, enabling the network to obtain more accurate detection capability.

Innovation:

Input side. The innovation here mainly refers to the improvement of input side during training, mainly including Mosaic data enhancement, cmBN and SAT self-confrontation training. Mosaic Data Enhancement method: mixing four images with different semantic information can enable the detector to detect targets beyond the normal context and enhance the robustness of the model. Since BN is calculated from four images, reliance on large mini-batch can be reduced. Self-Adversarial Training: Self-antagonistic training is a new data enhancement method, which can resist the antagonistic attack to

some extent. It consists of two stages, each stage carrying out one forward propagation and one back propagation

BackBone network. Combines various new ways, including CSPDarknet53, Mish activation function, and Dropblock [28].

Neck. Each of these sets of layers, such as the SPP module in Yolov4, the FPN [29] +PAN structure, is usually inserted between the BackBone and the final output layer.

Prediction. The anchor frame mechanism of the output layer is the same as Yolov3, which mainly improves the loss function CIOU_Loss [30] during training and the NMS filtered by the Prediction box changes to DIOU_nms [30].

2.5 YOLOv4-tiny (Compared with YOLOv4)

YOLOv4 is a kind of architecture specially designed for low-end GPU. It adopts CSPOSANet and PCB architecture to form the backbone of YOLOv4. No matter which hardware platform, YOLOv4-tiny can achieve real-time performance. Mish activation function was not used in feature extraction, and only one feature pyramid was used in feature enhancement layer, without further down-sampling like Yolov4.

3. Experiments

By comparing the performance of different Yolo models, we find a UAV real-time object detection algorithm with less trainable parameters and lower computational cost. We have proved the validity of Yolo model on the Pascal VOC dataset through experiments. The Yolo family model is based on the publicly available Darknet implementation. We used a Linux server with Intel (R) Xeon (R) silver 4110 CPU @ 2.10GHz, 64GB ram and an NVIDIA gtx2080ti GPU card to train and evaluate the model.

3.1 Datasets

Pascal VOC dataset [22] consists of 21493 static images in different places at a different angle. The training and validation sets contain 16551 and 4952 images, respectively. Images are labeled annotated with bounding boxes and twenty predefined classes (i.e., person, bird, cat, cow, dog, horse, sheep, aeroplane, bicycle, boat, bus, car, motorbike, train, bottle, chair, dining table, potted plant, sofa, and tv/monitor). All models in this paper are trained on the training set and evaluated on the validation set.

3.2 Training

Following the default configurations in Darknet, we train YOLOv3, YOLOv3-tiny, YOLOv3-SPP3, YOLOv4, and YOLOv4-tiny using SGD with the momentum of 0.9 and weight decay of 0.0005. We use an initial learning rate of 0.001 that is decayed by a factor of 10 at the iteration step of 70000 and 100000. We set the maximum training iteration as 120200 and use a mini-batch size of 64. We set the size of the input image as 416. Multiscale training is enabled by randomly rescaling the sizes of input images. We initialize the backbone networks of these models with the weights pre-trained on ImageNet.

3.3 Evaluation metrics

We evaluate all these models based on the following 4 metrics: (1) mean of average precision (mAP) measured at 0.5 intersections over union (IOU), (2) parameter size, (3) inference time as frames per second (FPS), (4) test time on validation sets. Specifically, the objectiveness confidence are set as 0.1 and non-maximum suppression threshold are set as 0.5 for all models in our experiments, respectively. We run an evaluation with no batch processing on one NVIDIA GTX2080ti GPU card using Darknet. Besides, we evaluate all models with input size as 416×416.

3.4 UAV Simulation Platform – XTDrone

XTDrone [23] is a UAV simulation platform based on PX4 and ROS (currently the simulator uses gazebo, and the connection with Airsim is under development). At present, it supports multi rotor aircraft (including four axis and six axis), fixed-wing aircraft, vertical takeoff and landing fixed wing aircraft

(including quadplane, tailsitter and tillrotor) and unmanned vehicle. The algorithm verified on XTDrone can be easily deployed to real UAV.

4. Results and Discussions

We compare the detection performance of all models on a validation set of Pascal VOC dataset in table 2.

Table 2. Evaluation results of all models.

	mAP	FPS	Parameters(M)	Test time(s)
YOLOv3	76.55	35	61.2	122.23
YOLOv3-tiny	62.5	134	8.7	28.14
YOLOv3-SPP3	76.87	40	63.9	128.19
YOLOv4	87.48	72	27.6	103.86
YOLOv4-tiny	50.06	252	7.2	18.41

As shown in Table 1, YOLOv4 achieves the best detection results, and the test time is reduced by 15.6% compared with YOLOv3 due to YOLOv4 improved the network and added many innovations. The training parameters of YOLOv4-tiny model are less than those of YOLOv3-tiny model, but the detection results are unsatisfactory, only reaching 50.06%, and the detection effect is the worst among all models. The performance of YOLOv3-SPP3 model is slightly better than that of YOLOv3 (due to the SPP module), but it is also far inferior to that of YOLOv4. These results show that if the training parameters and model size are not taken into account, the detection accuracy of YOLOv4 is the highest. However, to deploy on a UAV platform, the accuracy of YOLOv3 tiny is reduced, but the model is smaller, and the detection is faster, so it is more suitable for practical UAV platforms. We generate visual detection results of YOLOv4 and YOLOv3-tiny on challenging frames captured by our UAV, as shown in the figure. The two detectors can detect most of the objects in the frame accurately without significant difference.

5. Conclusion

This paper proposes to learn efficient deep object detectors by comparing current popular object detection models. We compare the performance of different Yolo series models on the Pascal VOC dataset, using mAP and FPS as evaluation metrics, and applying the training results to the XTDrone UAV Simulation Platform testing. Although the detection accuracy of YOLOv3-tiny is not up to the level of YOLOv4, it runs faster. As we all know, power consumption and failure are always positively correlated. Low power consumption is usually required by UAV applications to ensure the UAV's endurance. Therefore, for real-time UVA applications, YOLOv3-tiny is faster and better than the original YOLOv4. Pascal VOC dataset is a very challenging dataset with a high category imbalance. In our experiments, the problem of category imbalance has not been solved. A category with the largest number of instances might dominate the optimization of detectors. Future research should develop various approaches for solving the category imbalance problem in order to improve the detection accuracy of both baseline models and pruned models. With the development of object detection technology, we will continue to apply new object detection algorithms in the UAV field to find the optimal algorithm that meets the requirements of speed, accuracy, and network size at the same time.

References

[1]. M. Bhaskaranand, a. J. (n.d.). Low-complexity video encoding for UAV reconnaissance and surveillance. Proc. IEEE Military Communications Conference (MILCOM), pp. 1633-1638, 2011.

[2]. Blaschke, T.; Lang, S.; Hay, G. Object-Based Image Analysis: Apatial Concepts for Knowledge-Driven Remote Sensing Applications; Springer Science & Business Media: Heidelberg, Germany, 2008.

[3]. Dong, Q.; Zou, Q. Visual UAV detection method with online feature classifification. In Proceedings of the IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 December 2017; pp. 429–432.
[4]. Moranduzzo, T.; Melgani, F.; Bazi, Y.; Alajlan, N. A fast object detector based on high-order gradients and Gaussian process regression for UAV images. Int. J. Remote Sens. 2015, 10, 2713–2733.
[5]. Dong, Q.; Zou, Q. Visual UAV detection method with online feature classifification. In Proceedings of the IEEE 2nd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, 15–17 December 2017; pp. 429–432.
[6]. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015).
[7]. He, K., Zhang, X., Ren, S. & Sun, J. Deep Residual Learning for Image Recognition. in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 770–778 (IEEE, 2016). doi:10.1109/CVPR.2016.90.
[8]. Huang, G., Liu, Z., Maaten, L. van der & Weinberger, K. Q. Densely Connected Convolutional Networks. in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2261– 2269 (IEEE, 2017). doi:10.1109/CVPR.2017.243.
[9]. Ren, S., He, K., Girshick, R. & Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Trans. Pattern Anal. Mach. Intell. 39, 1137–1149 (2017).
[10]. Liu, W. et al. SSD: Single Shot MultiBox Detector. Preprint at https://arxiv.org/abs/1512.02325 (2016).
[11]. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. Preprint at https://arxiv.org/abs/1505.04597 (2015).
[12]. He, K., Gkioxari, G., Dollar, P. & Girshick, R. Mask R-CNN. in Proceedings of the IEEE international conference on computer vision 2961–2969 (2017).
[13]. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.
[14]. Shrivastava, A.; Gupta, A.; Girshick, R. Training region-based object detectors with online hard example mining. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 761–769.
[15]. Girshick, Ross. "Fast r-cnn." Proceedings of the IEEE international conference on computer vision. 2015.
[16]. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C. Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherland, 8–16 October 2016; pp. 21–37.
[17]. Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[18]. Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[19]. Redmon, J., and A. Farhadi. "YOLOv3: an incremental improvement (2018)." arXiv preprint arXiv:1804.02767 (1804). https://pjreddie.com/darknet/yolo/.
[20]. Victor, G.R.; Juan, A.R.; Jose, M.M.G.; Nuria, S.A.; Jose, M.L.M.; Federico, A. Automatic Change Detection System over Unmanned Aerial Vehicle Video Sequences Based on Convolutional Neural Networks. Sensors 2019, 19, 4484.
[21]. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. YOLOv4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.
[22]. Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. Int. J. Comput. Vision 2010, 2, 303–338.
[23]. He, Kaiming, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition." IEEE transactions on pattern analysis and machine intelligence 37.9 (2015): 1904-1916.

[24]. Ye J, Lu X, Lin Z, et al. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers[J]. arXiv preprint arXiv:1802.00124, 2018.
[25]. Liu, Zhuang, et al. "Learning efficient convolutional networks through network slimming." Proceedings of the IEEE International Conference on Computer Vision. 2017.
[26]. Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8759–8768, 2018.
[27]. Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu, Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. CSPNet: A new backbone that can enhance learning capability of cnn. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop), 2020.
[28]. Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. DropBlock: A regularization method for convolutional networks. In Advances in Neural Information Processing Systems (NIPS), pages 10727–10737, 2018.
[29]. Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, ´ Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2117–2125, 2017.
[30]. Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren. Distance-IoU Loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artifificial Intelligence (AAAI), 2020.