Cascade single stage Detector using full convolution network

Ming Liu, Zhengfa Yu, Kai Fu, Xiangming Zhou
ZHEJIANG DAHUA TECHNOLOGY CO., LTD, HangZhou, 310053, China
liu_ming@dahuatech.com yu_zhengfa@dahuatech.com

Abstract. For object detection, the two-stage approach always achieves better performance than one-stage. But one-stage is more effective. The main shortcomings of one-stage approach are: class imbalance problem during the training phase and sometimes the correct object score is lower. To address those problems, we propose a novel single-shot detector, called Cascade detection. It is composed of several stages, and each stage is composed of several detection layers. The main point of our method is: we design an anchors to anchors module (A2AM), which gives one-stage approach the ability of training stage by stage. It can not only address the class imbalance problem in the training phase, but also increase the object scores and the IoU between prediction and ground truth in the inference phase. The multitask loss function enables us to train the whole network in an end-to-end way. Experiments on PASCAL VOC 2007, PASCAL VOC 2012 and MS COCO prove that our method achieves state-of-art performance detection accuracy and computation efficiency.

1. Introduction
Object detection plays an important role in the computer vision community. It lays a foundation for computer vision applications such as video surveillance, face analysis, autopilot. Object detection has achieved high performances in recent years, thanks to the advanced deep convolutional neural networks.

Almost all the object detection benchmarks use PR-curve to evaluate the mAP of detector. So that when a detector has both high level of recall and precision, it is a good detector. The main challenge of object detection is having both high level of recall and precision. By doing the experiments, we find that we can separate object detection by two steps: location regression and classification. The current CNN detectors of state-of-art can be divided into two types: the one-stage approach, and the two-stage approach. Both approaches use the concept of anchor boxes. Two-stage approach generates region proposals by the preset anchors and then do the classification and location regression. The performances of two-stage approach are top on several challenging benchmarks.

The one-stage approaches [8, 16, 26] design the anchor boxes (default boxes) on different features maps with multiple scales and aspect ratios. This approach directly does the location regression on the anchor boxes, and classifies them whether an object, then classifies them into specific classes. The main advantage of doing this is high computational efficiency. But in general the mAP of one-stage detector is lower than two-stage detector. By doing the experiments, we found that object location regression is a simple task for all one-stage methods. The main reason of inaccuracy of object bounding box is the correct bounding boxes are filtered by score or suppressed by the non-maximum suppression. The reason for this is class imbalance problems. When the anchor boxes fit the ground truths, only very few anchor boxes satisfy the constraint. Most samples are easy negative examples which not contribute the networks learning. So that networks can’t classify the object correctly.
In recent years, some one-stage methods aim to avoid the class imbalance problem. Kong et al. \cite{22} propose focal loss which is better than the standard cross entropy loss, focal loss adds a weighting factor to balance positive, negative samples, and add a modulating factor to balance the hard, easy samples. Refine-Det \cite{20} proposes 3 modules to solve this problem. Anchor refinement module (ARM) to filter negative anchors, Object detection module (ODM) to take the outputs of ARM as input to do the further location regression and classification, Transfer connection block (TCB) connect the previous two modules.

In this paper we propose a novel detector called Cascade detection, it is composed by two parts: (1) multi-stages module, (2) anchors to anchors module, the networks structure is shown in Figure 1. The popular one-stage detectors Yolo \cite{17}, SSD \cite{26}, use Feature Pyramid Network (FPN) \cite{23} to extract the multi-scales features, with the preset anchors to predict the objects. We cascade the scores of each anchor which generated by different stage, and do the multi-stages location regression, which makes the detection more accurate in complex scenes. Detection layers in each stage are similar to Yolo, SSD, the difference is we use several stages. Anchors to anchors module (A2AM) connects anchors in different stages, it filters some easy negative samples by each stage, to balance the positive, negative samples, and also generate prediction bounding boxes as preset anchors for next stage’s regression. And with multi-task loss function, we can train our network end-to-end. To compare to other one-approach detector, we use the same backbone networks such as VGG-16 \cite{21} and ResNet-101 \cite{10} as our backbone network. We do not use the more recent ResNeXt-101 \cite{19} as our backbone network, because we want to reduce the influence of powerful backbone, and to illustrate that adding our cascade detection structure out-performs the previous one-stage detector.

We do some experiments on PASCAL VOC 2007, PASCAL VOC 2012 \cite{12}, and MS COCO \cite{24} benchmarks. These experiments demonstrate that Cascade detection out-performs the one-stage state-of-the-art methods. For example, we use our modules on Yolov3 \cite{17}, it achieves 83.2% and 82.4% mAP on VOC 2007 and 2012, which is better than the original Yolov3. It achieves almost the same level of accuracy as two-stage detectors, but also keeps time effectiveness. It runs at 29.2 FPS at NVIDIA Titan-X GPU with 512 x 512 input sizes in inference.

The contributions of this work can be summarized as follows:

- Cascade detection structure can be applied to any one-stage detector. It can improve the accuracy of detector with small increments of computation, and with this structure we can train our network end-to-end.
- Cascade detection uses the A2AM to connect the anchors in different stages. In training phase, it can balance positive, negative samples. In inference phase, it cascades score, regresses the location of each anchor.
- Cascade detection achieves state-of-art performance on several benchmarks.

2. Related Work

In recent years, object detection has made great progress with the development of deep neural network. The mainstream target detection methods are divided into two modes: (1) Two-stage target detection method represented by rcnn \cite{15,4,18}. (2) One-stage target detection method represented by yolo \cite{8,16,17}, ssd \cite{26,2}. mAP is an important index to measure the performance of object detector. In this index, the two-stage object detection method is obviously better than the one-stage object detection method. It is important to improve the accuracy as far as possible for a high mAP under a certain recall rate. Although the computational cost of one-stage method is much lower than that of two-stage method, the recall rate and accuracy of one-stage method are often lower than that of two-stage method.

Two-Stage Approach. There are two steps in the two-stage approach. First, some sparse ROIs are generated, and then these ROIs are classified and corrected. R-cnn \cite{15} starts the first stage of two-stage detection using CNN network, and then fast r-cnn \cite{4} replaces the traditional selective search to extract ROIs from CNN network, which improves the computational efficiency and performance. In recent years, many two-stage target detection methods based on RCNN framework emerge in
endlessly, such as network structure design [3, 5, 28], training strategy [1, 27], redesign loss functions [6, 3].

One-Stage Approach. Because of the efficient computing performance and good detection effect, the one-stage detector has been widely used in industry. Over-Feat [14] is one of the earliest one-stage detectors for object detection using deep network. With the development of CNN network, some more famous algorithms Yolo [8, 16, 17], SSD [26] are emerging. Yolo algorithm is optimized three times, through a single CNN network, according to the preset anchors to predict the object. The SSD algorithm assigns anchors of different sizes to feature maps of different sizes so that each layer can predict objects of different sizes. DSSD [2] and RRC [9] add deconvolution module on the basis of SSD to increase text information for shallow feature maps, thus improving accuracy. Retinanet [22] proposes focal loss to reduce the imbalance between positive and negative samples. Refinedet [20] proposes anchor refinement module and object detection module, which improves the performance without changing the structure of SSD network.

Class Imbalance and Foreground Classification. The main reason why the performance of one-stage object detection is not as good as that of two-stage method is that: (1) The imbalance of positive and negative samples. (2) One or a few layers do not have enough contextual information to classify and grade the target correctly. Almost all one-stage object detection methods preset a large number of anchors. In the training process, only a few of these anchors can match ground truth. A large number of negative samples lead to slow network learning and cannot correctly learn the real ground truth information. Focal loss and Refine-Det optimize this problem in loss function and network design respectively. Receptive field is important to object detection. Yolov2 [16] only use one feature layers to predict objects. Yolov3 [17] improve it by use multi-features layers to predict, which promote the mAP. It proves that different feature layers have different receptive fields, those features can distinguish different objects.

3. Approach
Viola and Jones [25] propose a cascade classifier which is faster than the classical Adaboost classifier. Traditional object detector uses sliding-window as candidate, and then classify these candidates...
whether object or not. This cascade classifier is concatenated by several simple AdaBoost classifiers. We assume that an Adaboost classifier need 200 features to achieve 99% TPR and 1% FPR. But for 99.9% TPR and 50% FPR only need 10 features. If we connect 10 classifiers, we get equation 1:

\[
\text{TPR} = (0.999)^{10} = 99.0% \\
\text{FPR} = (0.5)^{10} = 0.1% 
\]

Inspired by Equation 1, we apply cascade structure into one-stage approach. The anchor box in CNN detection is similar to the sliding-window in traditional detection. In one-stage approach, we use our cascade structure to filter, classifier, regress the anchor box to get the better performance.

**Anchor to anchor module.** A2AM can map the anchors of detection layer groups on different stage of detection. This module can connect the detection-layers groups on different stages. In the phase of training, this module can remove the anchors which corresponding to negative samples on each stage. It mitigates the class imbalance problem. And also do the classification and regression stage by stage. We can use K-means to get the width and height of anchor box. And we arrange the boxes by the size of feature maps. See figure 2, an example of how to arrange the anchor box. Algorithm 1 shows the steps of A2AM.

**Algorithm 1**

The processing of A2AM.

A. **Require:**
Current anchors, feature maps size, ground truths;

B. **Ensure:**
Next stage anchors;

1: Mapping the positive and ignore anchors to next stage feature maps. If the feature map of current stage anchor has the same size as feature map of next stage anchor, we project the same location anchor as positive or ignore anchor. If the sizes are different, project the location as:

\[
x_n = \frac{x_c \times w_c}{w}, \\
y_n = \frac{y_c \times h_c}{h} 
\]

We also mapping the location of predictions as the next stage’s anchors to regress the coordinate of object the location (points, sizes) is below:

\[
x_n = x_c + w_c \cdot a_0 \\
y_n = y_c + h_c \cdot a_1 \\
w_n = w_c e^{a_2} \\
h_n = h_c e^{a_3} 
\]

2: We use the score of current stage to control the proportion of positive and negative samples of next stage. Assume that we have N positive samples, we can take top 2N negative samples;

The disadvantage of Yolo algorithm is that the accuracy of object position recognition is poor and the recall rate is low. The main reason is that the number of anchor is small, which can not fully mine the information on each feature map. We add A2AM module on the basis of Yolov3 network, which inherits the advantages of Yolov3 while improving its disadvantages. We put all 9 anchor boxes in the shallow layer to determine the initial location of the detection target, then map the large anchor and small anchor in the upper sampling layer of feature fusion and the middle layer respectively, and further determine the location of the large target and small target. Finally, the larger anchor boxes are mapped in the deep layer, the medium-sized anchor boxes are mapped to the first upper sampling layer, and the small anchor boxes are mapped to the second upper sampling layer. The FPN structure is also added to the up sampling to fuse the features of the deep layer and the shallow layer. By using A2AM module, small target, medium target and large target can be accurately located. In addition, we only do one anchor box traversal of the pixels, and on the subsequent feature map, we only make further
accuracy on the bounding box determined in the previous stage. The allocation of anchor boxes in each stage on the feature map is shown in Figure 2.

Figure 2. For the shallow stage’s feature map, we put all 9 anchors on each. And for the next two stage we put 5, 4, and 3, 3, 3 anchors on each feature map respectively.

Figure 3. How A2AM connect all the anchors

**Multi-stage module.** At present, the one-stage approach (Yolo [17], SSD [26], etc.) uses to predict the location (points and size) and classification of targets on multiple feature layers, but in some complex scenes such as overlapping, small object, there may be some effect problems. In order to address these problems, we propose a multi-stage detection method. In each stage we have several detection layers. Those detection layers in each group are similar to the detection layers in one-stage approach. The loss function and samples match strategy is same to Yolo, SSD. In our method, the difference from the one-stage approach is that the whole structure of the algorithm consists of multi-stages. But the anchor contained in each stage is the same. The main steps of multi-stage module are as follows:

1. From the shallow layer of the whole network, we pull the first stage. We distributed N anchor boxes on this stage’s detection layers. Through this stage, computing the IoU between predictions and GTs to determine that positive and negative anchors.
2. Repeat use A2AM to connect anchors in all stages, shown in Figure 3.
3. Finally, the anchor points of all stages are transformed into coordinates and corresponding scores, categories and other information.
4. Experiments

We do the experiments on three datasets to evaluate our method. Pascal 2007 and Pascal 2012 [12] datasets are the earliest dataset to evaluate the performance of a detector. Both datasets include 20 categories. These 20 categories cover the normal object in our life. MS COCO [24] is an enhanced version of PASCAL dataset. It has more images, and even more categories. 80 categories cover animal, transport, food, sports, etc. The pictures in COCO include natural pictures and common objects pictures in life. The background is complex, the number of objects is large, and the size of objects is smaller. Therefore, the tasks on the COCO dataset are more difficult. In training phase, we use some common tricks to train the networks. In the phase of inference we select different combinations of stage to get better results. And we also do the ablation studies to prove our module can improve the performance. At last, we compare our method to some state-of-arts.

4.1. Training and Inference

Data Augmentation. Data augmentation which is important to build a robust model. As the images number is limited in the dataset. Using data augmentation can make model more robust. We randomly expand and crop the original training images. It generates more different sizes objects and the location of objects has changed. Apply additional random photometric [26] distortion and random flipping on training image can enhance the diversity of samples.

Loss function. As we mentioned before the network is trained with the multi-task loss function. The loss for each stage of detections is independent. And for each stage, the loss function is same as one-stage approach’s loss function. We replace with GIou [6] loss to predict the coordinate of object. The difference between standard one-stage approach is that we only do classification task on final stage, so only final stage have confidence loss. The total loss of networks is:

\[ L_i = \sum_{i=1}^{N} L_i \]  

N is the number of the stage \( L_i \) is the loss of each stage, which is as:

\[ L_i = \begin{cases} \delta_i (\delta_y \sum L_{xy} + \delta_{wh} \sum L_{wh} + \delta_{conf} \sum L_{conf}) & i < N \\ \delta_i (\delta_y \sum L_{xy} + \delta_{wh} \sum L_{wh} + \delta_{conf} \sum L_{conf} + \delta_{cls} \sum L_{cls}) & i \geq N \end{cases} \] 

\( \delta_i \) indicates the specific weight for each loss, and we only do the categories classification in the last stage.

Training strategy. We use reduced VGG-16 [21] network and ResNet-101 [10] as our backbone network. Both networks are pretrained on the ImageNet dataset. We use the standard SGD optimizer, with 0.9 momentum and 0.0005 weight decay. At the beginning of training, we start with very low learning rate and increase to 10-3. And for the first stage of detection, we don’t care about categories classification, so we can set the confidence loss weights as zero for first stage.

Inference. Although cascade detection is combined by multi-stage detection, the main computation load is on backbone network and down-sample convolution layers set. There is not much increase in computation compare to the original one-stage approach. At inference phase, we compute the results stage by stage. We map the previous stage’s results as anchor boxes of current stage, and multiple the object scores of previous on current stage’s scores. At last stage, the coordinates is regressed by previous. And then, we do soft-nms [13] on the candidates to get the final results.

Table 1. Detection results on PASCAL VOC dataset.

| Method          | Backbone   | Input size     | FPS  | mAP VOC2007 | mAP VOC2012 |
|-----------------|------------|----------------|------|-------------|-------------|
| two-stage:      |            |                |      |             |             |
| Fast R-CNN [4]  | VGG-16     | ~1000 × 600    | 0.5  | 70.0        | 68.4        |
| Faster R-CNN [18]| VGG-16     | ~1000 × 600    | 7    | 73.2        | 70.4        |
| Faster R-CNN [18]| ResNet-101 | ~1000 × 600    | 2.4  | 76.4        | 73.8        |
Table 2. Detection results on MS COCO test-dev set.

| Method       | Backbone   | AP   | AP50  | AP75  | APS  | APM  | APL  |
|--------------|------------|------|-------|-------|------|------|------|
| two-stage:   |            |      |       |       |      |      |      |
| Faster R-CNN [4] | VGG-16     | 21.9 | 42.7  | —     | —    | —    | —    |
| Faster R-CNN++[18] | ResNet-101 | 34.9 | 55.7  | 37.4  | 15.6 | 38.7 | 50.9 |
| R-FCN [7]    | ResNet-101 | 29.9 | 51.9  | —     | —    | —    | —    |

| one-stage:   |            |      |       |       |      |      |      |
| YOLOV2 [16] | DarkNet-19 | 21.6 | 44.0  | 19.2  | —    | —    | —    |
| SSD321 [2]  | ResNet-101 | 28.0 | 45.4  | 29.3  | 6.2  | 28.3 | 49.3 |
| SSD300 [26] | VGG-16     | 25.1 | 43.1  | 25.8  | 5.6  | 25.9 | 41.4 |
| SSD512 [26] | VGG-16     | 28.8 | 48.5  | 30.3  | 10.9 | 31.8 | 43.5 |
| SSD513 [2]  | ResNet-101 | 31.2 | 50.4  | 33.3  | 10.2 | 34.5 | 49.8 |
| RefineDet512 [20] | ResNet-101 | 36.4 | 57.5  | 39.5  | 16.6 | 39.9 | 51.4 |
| YOLOV3416[17] | DarkNet-53 | —    | 55.3  | —     | —    | —    | —    |
| YOLOV3608[17] | DarkNet-53 | 33   | 57.9  | 34.4  | 18.3 | 35.4 | 41.9 |
| CascadeDet512 | ResNet-101 | 33.9 | 56.4  | 41.1  | 21.0 | 41.4 | 52.7 |
| CascadeDet512+ | ResNet-101 | 40.7 | 58.4  | 43.4  | 23.4 | 43.8 | 54.4 |

4.2. Ablation studies

We do a few ablation studies to show that each module is effective. A2AM is essential for Cascade detection, so for each study A2AM is necessary.

Number of Stages. Table 3 shows the influence of the number of stage. Zero stage is the original Yolov3 approach. And with the increase of stages the improvement of effect is obvious. It shows that three stages has the best performance. Adding the extra fourth stage, do not increase the performance as expected.

Table 3. The influence of number of Stages. All models are trained on VOC207 and VOC2012 trainval set. Anchors distribution is same for each experiment.

| Number of Stage | mAP | VOC2007 | VOC2012 |
|-----------------|-----|---------|---------|
| 1               | 78.9| 78.6    |
| 2               | 80.1| 79.1    |
| 3               | 81.4| 80.3    |
| 4               | 81.2| 89.5    |
Figure 4. Detection results of Cascade detection on the MS COCO test-dev set. The backbone net is ResNet-101, and we only use MS COCO trainval as training data.

4.3. Comparison to state-of-the-art
We evaluate CascadeDet on the PASCAL VOC 2007 2012 and MS COCO dataset. For PASCAL VOC 2007 and 2012, we only use VOC 2007 and VOC 2012 train-val sets as train data. No additional dataset is added to training data. As presented in Table 1, compare to Re-fineDet512 with same input size and backbone network, CascadeDet has better performance in both FPS and mAP. Compare to DSSD513, the mAP is almost the same, but our method is more effective. We do the multi-scale testing to evaluate Cascade detection and obtain 83.2% for VOC 2007, 82.4% for VOC 2012(CascadeDet512+). And for MS COCO we evaluate our method on the test-dev split and compare with other state-of-art methods. Results are shown in Table 2. Compare to RefineDet512, CascadeDet512 achieves 2.5% ap increase. With larger input size the performance is better. Figure 4 shows the detection results of our method. We set a score threshold of 0.5 for displaying. As shown in Figure 4, Cascade detection can detect small object(e.g. birds,bottle), and big object(e.g.train). Because of the A2AM, we control the positive, negative samples, the classification is more accurate.
5. Conclusion
In this paper, we design an Anchors to Anchors Module which allows us to train one-stage approach stage by stage. It addresses the class imbalance problem during the training phase. And with the stage by stage classification and location regression, the final object score and coordinate are more reliable. We only add fully convolutional layers, so the increment of computation is not heavy. We demonstrate it is effective and achieves state-of-the-art accuracy. In addition, our structure is universal, it can apply to any one-stage approach to improve performance.

References
[1] A. Gupta A. Shrivastava and R. B. Girshick. Training region-based object detectors with online hard example mining. CVPR, 2016. 3
[2] A. Ranga A. Tyagi C. Fu, W. Liu and A. C. Berg. DSSD: Deconvolutional single shot detector. arXiv, 2017. 2, 3, 5
[3] Z. Cai and N. Vasconcelos. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. arXiv, 2019. 3
[4] R. B. Girshick. Fast R-CNN. ICCV, 2015. 2, 3, 5
[5] S. Eum H. Lee and H. Kwon. Me r-cnn: multi-expert region-based cnn for object detection. ICCV, 2017. 3
[6] J. Gwak A. Sadeghian I. Reid H. Rezatofighi, N. Tsoi and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. CVPR, 2019. 3, 4
[7] K. He J. Dai, Y. Li and J. Sun. R-FCN: object detection via region-based fully convolutional networks. NIPS, 2016. 5
[8] R. B. Girshick J. Redmon, S. K. Divvala and A. Farhadi. You only look once: Unified, real-time object detection. CVPR, 2016. 1, 2, 3
[9] J. Liu W. Sun J. Pang Q. Yan Y.-W. Tai J. Ren, X. Chen and L. Xu. Accurate Single Stage Detector Using Recurrent Rolling Convolution. CVPR, 2017. 3
[10] S. Ren K. He, X. Zhang and J. Sun. Deep residual learning for image recognition. CVPR, 2016. 2, 4
[11] S. Ren K. He, X. Zhang and J. Sun. Deep residual learning for image recognition. CVPR, 2016. 5
[12] C. K. Williams J. Winn M. Everingham, L. Van Gool and A. Zisserman. The pascal visual object classes (voc) challenge. IJCV, 2010. 2, 4
[13] R. Chellappa N. Bodla, B. Singh and L. S. Davis. Soft-nms:improving object detection with one line of code. ICCV, 2017. 5
[14] X. Zhang M. Mathieu R. Fergus P. Sermanet, D. Eigen and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. ICLR, 2014. 3
[15] T. Darrell R. B. Girshick, J. Donahue and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR, 2014. 2
[16] J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. CVPR, 2017. 1, 2, 3, 5
[17] J. Redmon and A. Farhadi. Yolov3: An incremental improve-ment. arXiv, 2018. 1, 2, 3, 5
[18] R. B. Girshick S. Ren, K. He and J. Sun. Faster R-CNN: towards real-time object detection with region proposal net-works. IEEE Transactions on Pattern Analysis Machine In-telligence, 39(6):1137–1149, 2015. 2, 5
[19] P. Dollar Z. Tu S. Xie, R. B. Girshick and K. He. Aggregated residual transformations for deep neural networks. CVPR, 2017. 2
[20] X. Bian Z. Lei S. Zhang, L. Wen and S. Z. Li. Single-Shot Refinement Neural Network for Object Detection. CVPR, 2018. 1, 3, 5
[21] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv, 2014. 2, 4
[22] R. B. Girshick K. He T. Lin, P. Goyal and P. Doll’ar. Focal loss for dense object detection. ICCV, 2017. 1, 3
[23] R. B. Girshick K. He B. Hariharan T. Lin, P. Dollar and S. J. Belongie. Feature pyramid networks for object detection. CVPR, 2017.
[24] S. J. Belongie J. Hays P. Perona D. Ramanan-P. Doll’ar T. Lin,
[25] M. Maire and C. L. Zitnick. Microsoft coco: common objects in context. ECCV, 2014.
[26] P. A. Viola and M. J. Joness. Rapid object detection using a boosted cascade of simple features. CVPR, 2001.
[27] D. Erhan C. Szegedy S. E. Reed C. Fu W. Liu, D. Anguelov and A. C. Berg. SSD:single shot multibox detector. ECCV, 2016.
[28] A. Shrivastava X. Wang and A. Gupta. A-fast-rccn: Hard positive generation via adversary for object detection. CVPR, 2017.
[29] J. Wang X. Zhao Y. Wu Y. Zhu, C. Zhao and H. Lu. Cou-plenet: Coupling global structure with local parts for object detection. ICCV, 2017.