# Aspect Ratio Sensitive Network

**Project Report of CS272 Computer Vision**

Jianxiong Cai
SIST
ShanghaiTech University
caijx@shanghaitech.edu.cn

Second Author
Institution2
First line of institution2 address
secondauthor@i2.org

## Abstract

*Many objects in real world have a prior knowledge, which could be helpful for object detection. In this project, we propose an approach to include aspect ratio as the prior knowledge. Our approach duplicate the RPN network in faster RCNN [13] so that different RPN can focus on generating proposal of different shapes.*

*Our approach dramatically improves the performance (AP) of certain classes. However, our approach bring two major drawback: 1) the speed descreases as adding more RPN. 2) more false negative appear as different RPNs generate equal number of proposal for now.*

## 1. Introduction

General-purpose object detection on RGB images plays an important role in various applications, like autonomous driving and security. In this project, we propose a new approach to include aspect ratio as the prior knowledge for object detection. Although aspect ratio is a relatively weak feature for objects, we observes that it holds for certain object classes in most major datasets. Our approach dramatically improve the performance (AP) for those certain classes.

### 1.1. Aspect Ratio (Motivation)

As objects usually do not deform so much, they are generally with relatively fixed shape in some extent, which infers that the target bounding box will have certain relationship with aspect ratio especially for non-living objects. An intuitive example is that bottle will be more likely to have large aspect ratios (width to height), i.e. tall and thin. According to the statistics, both of the two commonly used datasets, VOC and MS COCO, are showing the regulation. Following histograms describe that for some classes objects are tend to be with some aspect ratios.

The original RPN network in Faster RCNN [13] have 3 anchor shape: 1:2, 1:1 and 2:1. Thus we count the distribution of object shape by dividing their aspect ratios into those three ratios, as is shown in fig.1 and fig.2.

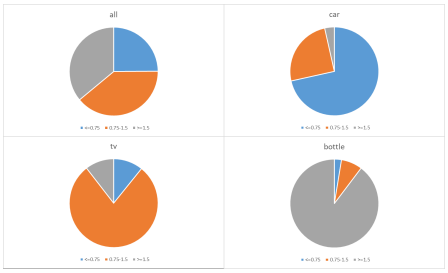


Figure 1. Pie charts of VOC2007

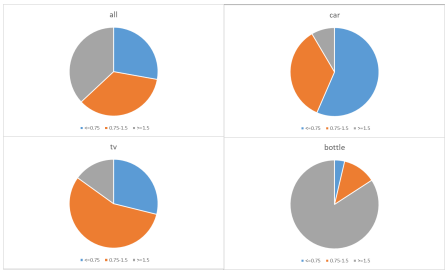


Figure 2. Pie charts of MS COCO

## 2. Related Work

### 2.1. R-CNN

[5] addressed the R-CNN detector which promote the accuracy of nueral networks for object detection. This R-CNN detector combines a region proposal part (Selective Search [14]), a CNN feature extractor and a classifer. R-CNN detector surmount conventional detectors in the aspect of accuracy but its speed is limited by generating massive region proposals and extracting features using CNN on each of the region proposal.

Spatial pyramid pooling (SPP) [7] introduced a way to meliorate by employ pyramid pooling layer. SPP-net speed up the detector by computing CNN only once per image.

Imitating SPP-net, Fast-RCNN [4] shown that using multi-task learning and back-propagation through ROI pooling layer will speed up the R-CNN detector by an order of magnitude.

The bottleneck of Fast-RCNN is conventional region proposal generator, so Faster-RCNN [13] proposed Region Proposal Network (RPN) which sharing features with classifer network.

## 2.2. Region Proposal

Most traditional region proposal methods are based on low-level features. Faster-RCNN introduced Region Proposal Network (RPN) to generate region proposals based on high-level semantic features. RPN surpass conventional region proposal methods, either unsupervised method (e.g. Selective Search [14] and EdgeBoxes [20]) or supervised method (e.g. BING [2]), on both speed and accuracy.

There are several approach of refining region proposals.

### 2.2.1 cascade refinement

Region Proposal Network can be refined by applying a multi-stage cascading pipeline structure. Applying a two-class Fast RCNN to refine proposals has shown significant improvement. [16] [18] [8].

DeepProposal [3] introduce a Cascading Deep Convolutional Layers to get feature maps. CascadedCNN [6] proposes a RefineNet added after RPN which can effectively reduce the number of proposals and improve their confidence. MTCNN [15] apply similar approach, using a R-net to refine P-net.

[9] introduce a spatial correlation related method, using a new EM-like group recursive learing approach to iteratively refine object proposals and provide an optimal spatial configuration of object detections.

### 2.2.2 multi-scale refinment

Methods like SDP,SSH or MS-CNN [17] [12] [1]make independent prodictions at different layers, which will ensure smaller objects are trained on higher resolution layers.

Other methods like FPN, RetinaNet [10] [11] propose a pyramidal structure to combine shallow layer with deep layer to get both high-level feature and low-level feature.

## 2.3. ROI and NMS

CoupleNet [19] propose a two branch network of ROI after RPN to obtain both local part information and global imformation.

# 3. Aspect Ratio Sensitive Object Detection

## 3.1. Aspect Ratio Sensitive Network

We have count the sapect ratio of different classes on different dataset. All datasets contain certain classes whose aspect ratio show obvious uneven distribution among different shapes.

## 3.2. Network Archiure

**ARS Net** To include aspect ratio as a prior knowledge for object detection, we modified the network archiure by duplicating the region proposal network into 3 copies, each handling objects of different shapes. During training (see figure 3), green rpn (right) is only provided with ground truth bounding boxes of aspect ratio greater than 2:1, grey one (middle) with boxes around 1:1 only and yellow one (left) with boxes smaller than 1:2 only. Results of three region proposal networks are simply stacked together in testing time. In this way, our network is able to learn separate filters for objects of different shapes. Non-maximum suppression is applied in the end to reject high over-lapping objects.

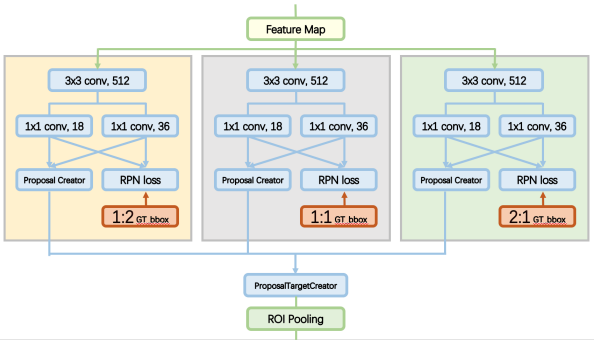


Figure 3. Multiple RPN approach

**Sharing Features** The multi-rpn network works, but comes at great cost. One of the obvious shortcomings of this approach would be the high overall time complexity. Compared with the original faster RCNN, which runs at about 9fps on one Nvidia TitanX, our multi-rpn approach can only run at half speed. Sharing features, plus adding additional one-by-one convolutional layers are applied to cope with such problem. More specifically (see Figure 4), three rpns share the same 3*3*512 layer, but each would have another 1*1*256 layer of its own to further reduce depth. Other parts of the network remains unchangde. From table 1, adding 1*1 conv into our network improves speed while maintaining a similar mAp. However, it's still slow when compared with the origin RCNN, afterall, region proposal network still remains to be the biggest overhead of faster RCNN.

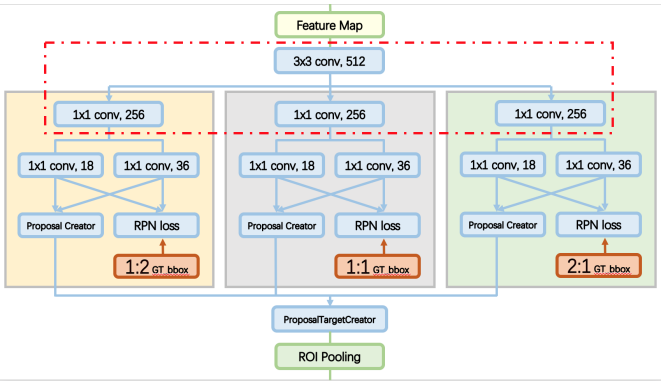


Figure 4. Sharing Features

| Approach | Frame per second (fps) |
|---|---|
| ARSnet (300 proposals) | 4.73 |
| ARSnet (900 proposals) | 3.56 |
| ARSnet+1*1 conv (300 proposals) | 5.04 |
| ARSnet+1*1 conv (900 proposals) | 3.84 |
| faster RCNN (300 proposals) | 9 |

Table 1. Time complexity under different settings reported on Nvidia TitanX. Adding 1*1 conv into our network improves speed while maintaining a similar mAp.

### 3.3. Training Details

In general, we adopted the training approaches from the origin faster RCNN. Fisrt 30 feature extraction layers are initialized with a pretrained VGG-16 model from caffe, other convolutional layers are drawed from a zero-mean Gaussian distribution with standard deviation with 0.01 (*REF FAST RCNN*). One single image is chosen randomly at each step to provide positive and negative samples up to 1:3 ratio for both rpns and final classifiers. We trained the network with stochastic gradient descent (SGD) end-to-end, learning rate is set to 0.001 for the first 15 epochs. Then we fine-tuned the model with the best accuracy with another learning rate of 0.0001 for the next 15 epochs. Our approach shows an significant increase of aspect ratio sensitive class.

## 4. Experiment

### 4.1. Experiments on PASCAL VOC

We evaluate our method on the PASCAL VOC 2007 detection benchmark. This dataset consists of 5011 train images and 4952 test images over 20 object categories. Evaluation is perforated on all test images. The metric we use is mean Average Precision (mAP). This is also a general metric in the field of computer vision. (How about recall)

We applied our method to Faster R-CNN with the public VGG-16 model, which has 13 convolutional layers and 3 fully-connected layers. Table 3.3 shows the results on VGG-16 model. Using ARS Net, the result is 73.2% for shared features between proposal and detection, compared to 71.8% that evaluated on single RPN. As shown above, our method has also dramatically improved AP in some categories such as bottle (by 3.9%) and cow (by 5.9%). This is because we specify each RPN processing selected ground truth during training, which leads to the aspect ratio sensitivity. Thus the proposals generated by ARS Net for those categories are more accurate. There's a slight decrease in several classes, which is a side effect. One possible reason is that the distribution of the test set is not consistent with the training set.

### 4.2. Number of Proposals

We further tested ARS Net with diverse number of proposals, which is as shown in table 4.2. When the number of proposals drops from 900 (Top 300 proposals are selected from each scale) to 300, mAP reduce by 1.7%. And a further decrease on number of proposals leads to dropping on mAP. The main reason we thought might be that more proposals means we may have multiple proposals for a single object, where the best proposal will be selected during none maximum suppression by score. In fact, when number of proposals is 1200 in total (400 each), mAP gets a slight increase (From 73.2% to 73.3%). However, this change dose more harm than good. It will lead to a decrease in fps. So we take 900 proposals in total as our final result. This maintains a relatively high mAP while also taking speed into account. After using the trick of sharing features, we make ARSNet as fast as original Faster R-CNN.

## 5. Discussion

### 5.1. Failed Tries

**Add non-maximum suppression(NMS) after the three RPNs.** Three RPNs means more region proposal and lead more false positives in our detection results as for original faster R-CNN will decrease some RoIs in the same locations with different shapes by NMS in RPN( it might be right for some cases but not always).

We simply apply NMS without score and lead mAP decrease by 0.5 roughly, decrease by 1 approximately when choose 300 RoIs after NMS. To avoid the influence caused by order, we shuffle the RoIs after stack them all together. Removing shuffle will make mAP down much more.

When apply NMS with score, we face a trouble that the three RPNs are score the RoIs independently so its hard to make comparison and there is also a question that if the scores should be normalized so we didnt implement the NMS with score in this step.

**Rescoring based on prior aspect ratio distribution.** The idea to reduce score of bounding boxes with rare aspect ra-

Table 2. Comparison

| method | # of proposals | data | mAP | areo | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Faster RCNN | 300 | VOC07 | 71.8 | 73.5 | 81.5 | 68.5 | 53.7 | 52.3 | 80.7 | 85.3 | 84.3 | 52.5 | 76.8 | 71.5 | 81.3 | 84.9 | 75.1 | 79.6 | 44.7 | 72.5 | 65.9 | 79.8 | 72.3 |
| ARSNet | 900 | VOC07 | 73.2 | 74.5 | 81.5 | 71.1 | 56.7 | 56.2 | 83.6 | 85.8 | 87.4 | 51.4 | 82.7 | 68.9 | 83.8 | 86.5 | 77.1 | 81.0 | 43.9 | 73.0 | 66.4 | 79.7 | 72.6 |

Table 3. Decreasing number of proposals in ARSNet

| method | # of proposals | mAP |
|---|---|---|
| ARSNet | 300 | 71.5 |
| ARSNet | 900 | 73.2 |
| ARSNet | 1200 | 73.3 |

tios is naive. We simply multiply predicted score of the probability(calculate from the statistics of dataset) of the aspect ratio occurred in its corresponding class. It leads more than 10 decrease although the thought seems to make sense at first glance.

We try to multiply the predicted score with the prior possibility and a constant k but it doesn't work as well.

It failed mainly because it is almost impossible to design a rescore mapping by hand and it is not robust for extreme cases. What's more, the rightness of prior possibility is questioning.

**Cascade head part.** All the tries can be summarized as duplicate removal indeed. The initial idea is to reshape the bounding boxes generated in the first head by prior ratio distribution to convey the class imformation to next head but fail to design a suitable reshape mapping based on aspect ratio. Then we try cascade purely but maybe some faults in programming or mistakes in traning lead that cascade method failed during training and it actually doesn't relate much about aspect ratio.

We try to decrease IoU of NMS and increase the lower bound of score when implementing the second head but failed. Additionally we also try to weight the loss of different head to make the second head a more accurate classifier and regressor but fail while training.(The loss function diverges at the first epoch and then we reduce the learning rate but the mAP of the first epoch is below 10 so we quit).

**Design new ways to do non-maximum suppression in the final suppression.** TBA.

## 6. Conclusion

By including aspect ratio as a prior knowledge, our network dramatically improves performance on detecting aspect-ratio sensitive objects. However, on other object, the performance reduce because of more false positive.

## References


[1] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. *CoRR*, abs/1607.07155, 2016.

[2] M. M. Cheng, Z. Zhang, W. Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293, June 2014.

[3] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, and L. V. Gool. Deepproposal: Hunting objects by cascading deep convolutional layers. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2578–2586, Dec 2015.

[4] R. Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, Dec 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, Washington, DC, USA, 2014. IEEE Computer Society.

[6] Y. Guo, X. Guo, Z. Jiang, and Y. Zhou. Cascaded convolutional neural networks for object detection. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4, Dec 2017.

[7] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1904–1916, Sept 2015.

[8] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2479–2487, Dec 2015.

[9] J. Li, X. Liang, J. Li, Y. Wei, T. Xu, J. Feng, and S. Yan. Multi-stage object detection with group recursive learning. *IEEE Transactions on Multimedia*, pages 1–1, 2017.

[10] T. Y. Lin, P. Dollr, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, July 2017.

[11] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, Oct 2017.

[12] M. Najibi, P. Samangouei, R. Chellappa, and L. S. Davis. Ssh: Single stage headless face detector. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4885–4894, Oct 2017.

[13] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017.

[14] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *2011 International Conference on Computer Vision*, pages 1879–1886, Nov 2011.

[15] J. Xiang and G. Zhu. Joint face detection and facial expression recognition with mtcnn. In *2017 4th International Conference on Information Science and Control Engineering (ICISCE)*, pages 424–427, July 2017.

[16] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Craft objects from images. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6043–6051, June 2016.

[17] F. Yang, W. Choi, and Y. Lin. Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2129–2137, June 2016.

[18] X. Zeng, W. Ouyang, J. Yan, H. Li, T. Xiao, K. Wang, Y. Liu, Y. Zhou, B. Yang, Z. Wang, H. Zhou, and X. Wang. Crafting gbd-net for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2018.

[19] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu. Couplenet: Coupling global structure with local parts for object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4146–4154, Oct 2017.

[20] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 391–405, Cham, 2014. Springer International Publishing.