IOP Conference Series: Materials Science and Engineering

Accurate object classification and detection by faster-RCNN

To cite this article: M Lokanath *et al* 2017 *IOP Conf. Ser.: Mater. Sci. Eng.* **263** 052028

Accurate object classification and detection by faster-RCNN

Lokanath M, Sai Kumar K and Sanath Keerthi E
School of Electronics Engineering, VIT University, Vellore, Tamil Nadu 632014, India

E-mail: lokanath.m@vit.ac.in

Abstract. Recent advances in object detection algorithms include fast and faster RCNN which made the detection times comparatively low with high accuracy. In this work we verify the integrity of a proposed algorithm which uses RPN (Region proposal networks) and Fast RCNN (Region based Convolutional Neural Networks) for the detection. The RPN provides region proposals from that we give the ROI (Regions of Interest) as input to the RCNN network, it can be further merged into a single network by sharing their convolutional features to detect a specific object in a given image. As we use a unified network there is no need to get the ROI from an external network which makes this process cost free. We trained VGGnet with two different data sets PASCAL VOC 2012 and MS COCO on a low cost GPU and verified the accuracies while comparing the outputs with increasing number of region proposals. As we increased the number of proposals we observed a significant increase in the mAP (Mean Average Precision) value till 2000 proposals from where it reached saturation. Our results are compared with the state of the art algorithm with an increase of 1.2% in terms of mAP for 1800 proposals.

1. Introduction

Object recognition is one of the most important part in computer vision. Object recognition initially started with digital image processing methods such as Edge detection, Recognition by parts, Gradient matching. With the recent advancements and introduction of deep neural networks object recognition has been more accurate and can be applied in real time with faster implementation. With the increase in the number of convolutional layers the detection is more accurate.

Generally detection and classification are done as two different steps RPN[1] is used for detection and RCNN [2] is used for classification. The RPN networks which provide us with the ROI (Regions of Interest) are to be done as an external process and these regions are shared with Faster RCNN. Some of these networks include Selective search, greedy merges, etc. Nevertheless, the region proposal step still consumes almost the same amount of time as the detection network. The two processes can be expensive on their own, but a cost effective way is to share the convolutions between. In this project we merge the RPN [1] and the RCNN [2] parts into a unified process which makes the process much easier. This can be an effective solution for better accuracy and detection time tradeoffs.

RPN networks are constructed by adding convolutional layers on top of the convolutional feature maps that are used by detectors like faster RCNN. By doing so, we create a fully connected network, which helps us generate region proposals for detection. This can be trained similar to the detection algorithms. As we consider multiple classed images we need the region proposals to be distinctive with a wide variety of aspect ratios and scales, unlike the existing methods that use different fixed sized filters over the image we use “anchor” boxes that help us get the required regions [3].

The anchor boxes have multiple references at which boxes of different aspect ratio and sizes are taken into consideration at once. The training process includes alternative fine tuning of region proposal (RPN) and detection (Faster RCNN) and converges them into one network to work on which gives the detected output. By doing so, we not only decrease the detection time, but also overcome the computational problems of the selective search algorithm. As far as the deep neural networks go we get better features based on the number of layers of the network being used. So, we considered VGGnet on both PASCL VOC 2012 and MS COCO and achieved results as per the mAP[3-4].

2. Related Work

Deep neural networks have gained much popularity starting from the Alexnet an 8 layered convolutional neural network that achieved good mAP scores. Considering the CNN [5] we have an image window that predicts the background and foreground pixels for the whole object including its top, bottom and side halves. The so predicted masks are then regressed by bounding boxes. But in the Faster RCNN algorithm we use these region proposals to classify and detect objects.

2.1 Neural networks for Object detection

RCNN is a visual object detection system that combines bottom up region proposals with features computed by the Conv neural network. RCNN uses a selective search method in which region proposals are taken from an external network

2.2 Object proposal methods

Some of the Object proposal methods include Selective search, Greedy suppression methods, constrained parametric min cuts etc. DPM [6] detectors can be used to propose the ROI. But as the object categories increase it is difficult for the DPM detectors to detect different classes with high accuracy. As we consider a unified network and merge the object proposal method we do not take external proposal methods into account instead we use the Over feat method in which we train the fully connected layer to give the box references for a single class. RPN on the other hand, uses a fully connected layer which is a convolutional layer to get the multiple class objects simultaneously by sharing all the computations of multiple classes which solves the problem with DPM. The so generated regions are class agnostic and this sums up for the RPN module which gives us the ROI which are shared with the Fast RCNN network for the detection of the objects from the ROI.

3. Object Detection using Faster RCNN

Faster RCNN is just a modified version of Fast RCNN which combines a RPN instead of the external region proposal network

3.1 Fast RCNN

Fast RCNN [7] is introduced as a refined network to overcome the repeated computing of convolutional features. The most important factor of Fast RCNN is that it shares the computation. After the region proposal, we'll get some regions which are bounded with a specific box regression. In the RCNN algorithm, they just directly feed the warped image to the CNN. That is, if we have 2000 proposals, we have to do 2000 times forward pass, which wastes lots of time.

Actually, we can use the relation between these proposals. Many proposals have overlap with others, and these overlap part is fed into the CNN for many times. But in Fast RCNN we can just compute them for once. As the convolutional layer does not change the spatial relation between the adjacent pixels. This can be used to project the coordinates in raw image to the corresponding neuron in the Conv layer and so we can compute the whole image through the CNN only once. After getting the feature for each bounding box we can give it as an input to the ROI pooling layer which is nothing but an SPP (Spatial Pyramid Pooling) layer which is then fed to a fully connected layer which in turn is divided into classes and bounding box regression layers [8-9].

3.2 Region Proposal Network

Faster RCNN replaces the external region proposals from selective search or any other proposal algorithm with a merge Region proposal network within the Fast RCNN. Usually RPN takes an image feature map as input, outputs a set of rectangular object proposals, each with an objectness score. We map each sliding window of n*n (considering n=3) onto a lower dimensional feature. The actual receptive field is quite large if you project the coordinate back to the raw input size. This operation is done by applying a 3*3*256 convolutional kernel on the feature map. In this way, we will get an intermediate layer in 256-d. Then the intermediate layer will feed into two different branches, one for an objectness score (classifier) and the other for regression.

Each of the sliding window [10] has an anchor centered with different sizes and aspect ratios, so we have a total of W*H**k* anchors where W and H are width and height of the anchors respectively and *k* is the number of anchors. These anchors provide us a cost efficient way to get the region proposals by using a pyramid approach.

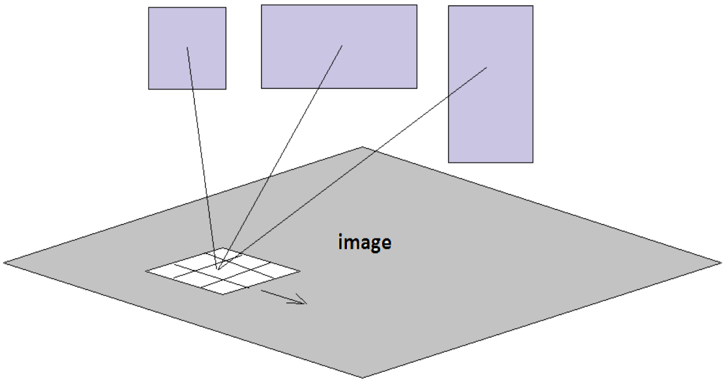


Figure 1. Multiple references of anchor boxes applied to the image with ratios 1:1, 1:2, 2:1 with a window of 3*3 sliding over the image.

4. Implementation

4.1 Training

Training of the RPN network requires updating the weight and reduce the loss function.

4.2 Weight Initialization

Once the Network gets trained the final value of all weights are unknon to us, but we can assume symmetry and ppproximate half of the weights “will be positive and half of weights will be negative after proper data normalization. As the “best guess” we do by reasonable sounding is to initialize all weights as zero’s. But there is no source of asymmetry between neurons if their weights are initialized to be same. As symmetry breaking we want to initialize the weight of neurons with small numbers. The whole thinking is that neurons are unique and behave randomly in the starting, as the training progress they will update distinctly and fit themselves as diverse parts of the entire network.

4.3 Loss function

Loss function measures the compatibility between the prediction and the ground truth label. We assign a positive label based on IoU (Intersection over Union) [11] overlap, i.e., A positive label for anchor with highest IoU overlap or with a value greater than 0.7 with a ground truth label. Similarly, for non-positive labels we assign a negative label if the IoU overlap is less than 0.3 for all ground truth labels. And the loss function can be given as:

$$L\ (\{p_i\}, \{\ t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}\ (p_i, p_i^*) + \gamma \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}\ (t_i, t_i^*)$$

Here i is the index of the anchor, p_i is the probability of anchor is an object, p_i^* is the ground truth label, i.e., It is 1 if positive label and 0 for negative anchor, *it* represents the bounding box coordinates of the anchor, t_i^* refers to ground truth box, L_{cls} is log loss, N_{cls} and N_{reg} are the normalization values with values 256 and 2400 respectively. But we can balance regression and classifier by multiplying the regression part by a factor (γ) =10.

When training [12] the RPN we consider only 256 anchors such that the positive and negative labels are almost equal and calculate the loss function. If we don't have a considerable amount of positive samples we pad them with negative ones.

The feature sharing is done by alternating training of features between the RPN and RCNN networks During implementation, we consider 3 different anchor ratios of 1:1, 1:2, 2:1. And around 8000 anchors are considered during the training process, but by fixing IoU to 0.7 we decrease these proposals to 2000 approximately by taking the top 2k proposals and only these are used to train RCNN network.

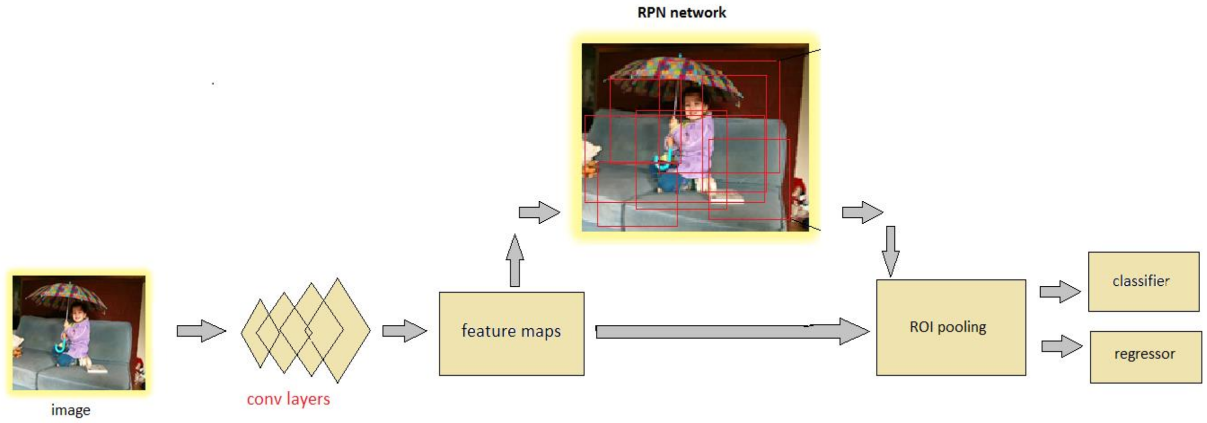


Figure 2. An overview of the Faster RCNN network with RPN module included as a unified network within the whole flow

5. Datasets

We considered 2 datasets namely PASCAL VOC 2012 and MS COCO for training and testing parts.

PASCAL VOC 2012:

This data set [13] contains 20 object categories with about 5000 trainval image and 5000 images for testing. We train this on VGGnet which has 13 convolutional and 3 fully connected layers. We use steps of 300 proposals starting from 300 to 1500 object proposals during detection.

MS COCO:

This data set contains around 80k images for training and validation and 40k images in the testing set. We use VGGnet training on this set and evaluate the performance on the specified metrics given in the official MS COCO website [14]. The output data for calculating metrics for valuations in detections are taken in the specific format mentioned in the dataset website.

Figure 3. Example images from the MS COCO dataset Eggs, dining table, person, airplane, zebra, birds, pizza, bowl.

6. Results

The following are the results for the trained and tested networks VGGnet on MS COCO for different region proposals:

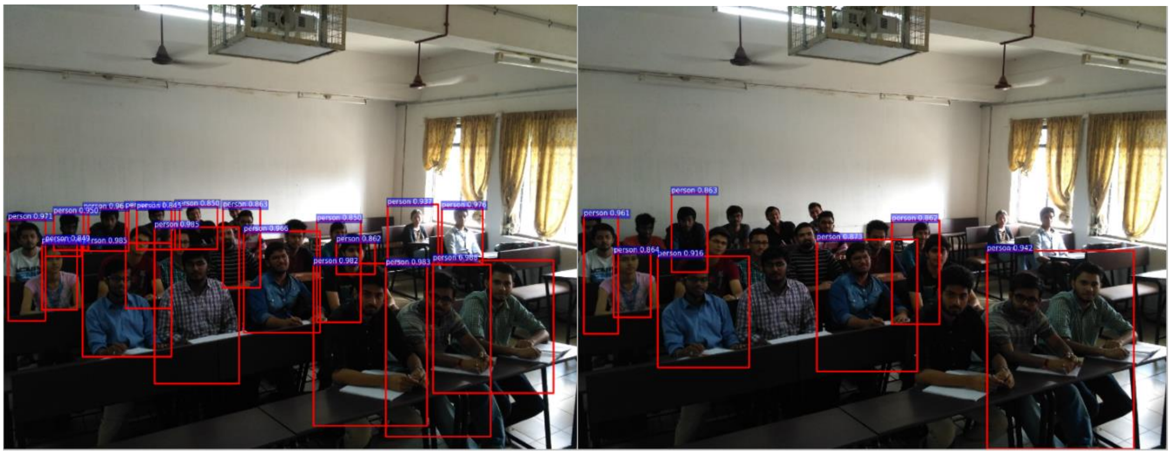
Table 1. mAP values for different number of region proposals

Metrics	Coco VGG 1800 Proposals	Coco VGG 900 Proposals	Coco VGG 300 Proposals*
AP.50	0.449	0.432	0.453
AP.75	0.253	0.283	0.234
APS	0.074	0.070	0.072
APM	0.271	0.254	0.264
APL	0.390	0.382	0.369
AR1	0.249	0.241	0.238
AR10	0.357	0.338	0.341
AR100	0.35	0.331	0.347
ARS	0.138	0.121	0.115

ARM	0.374	0.346	0.389
ARL	0.548	0.530	0.544
AP	**<u>0.254</u>**	0.244	0.241

AP=Average Precision, AR=Average Recall, AP.50=Average Precision at IoU= 0.5, AP.75=Average Precision at IoU=0.75, APS= Average Precision over small objects, APM= Average Precision over medium objects, APL= Average Precision over large objects, proposals*=values from Faster RCNN on MS COCO from reference paper [1].

With a considerable increase in the AP [15] values resulting from different proposals we get a better accuracy in the images. The following are the example images we performed our testing on:



4(a) Region proposal 1800: 4(b) Region proposals 300:

Figure 4. A picture of class room in VIT university, Comparing the two figures above, we can clearly observe the number of people detected in both cases. In the first case 4(a) with higher number of region proposals more objects are detected compared to the later one 4(b).

The following is the graph of mAP values of [16],[17] VGGnet at 1800 proposals on [18] PASCAL VOC 2012 mAP values of each object class:

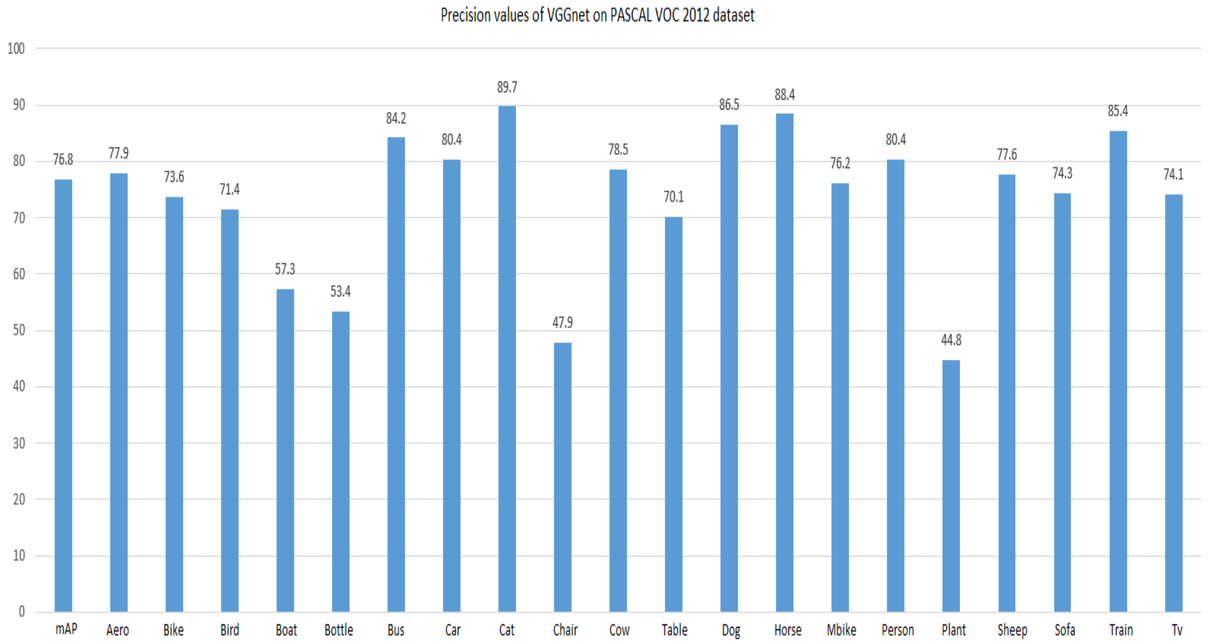


Figure 5. AP values for VGGnet on PASCAL VOC 2012 dataset

7. Conclusion

Considering the region proposals number we trade off the detection times, i.e., with higher accuracy we get more detection times compared to lesser region proposal detections. However, we were unable to achieve real time frame rates by increasing the region proposals. But there is not much change observed in the single object images even when there is a change in the number of region proposals in terms of visible accuracy compared to multiple object (of the same class) images.

References

[1] Ross Girshick, Jeff Donahue, Trevor Darrell and Jitendra Malik, 2016 Region-Based Convolutional Networks for Accurate Object Detection and Segmentation *IEEE Transactions On Pattern Analysis And Machine Intelligence* **38** 1 142-58

[2] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov 2014 Scalable object detection using deep neural networks *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 2155–62

[3] R. Girshick, 2015 Fast R-CNN *arXiv e-prints vol. arXiv:1504.08083* **1** [cs.CV]

[4] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, 2013 Pedestrian detection with unsupervised multi-stage feature learning *Proc. IEEE Conf. Comput. Vis. Pattern Recog* 3626–33

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman 2010 The PASCAL visual object classes challenge *Int. J. Comput. Vis* **80** 2 0303–38

[6] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Thor 2014 BING: Binarized normed gradients for objectness estimation at 300fps *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* 3286–93

[7] K. E. van de Sande, C. G. Snook, and A. W. Smeulders, Fisher and Vlad with flair 2014 *Proc. IEEE Conf. Comput. Vis. Pattern Recog* 2377–84

[8] Y Bangui, A Courville and P Vincen 2013 Representation learning: A review and new perspectives *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 8 1798–1828

[9] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei 2012 Imagenet large scale visual recognition competition (ILSVRC2012) [Online]. Available: *http://www.image-net.org/ challenges/LSVRC/2012*

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, 2014 Rich feature hierarchies for accurate object detection and semantic segmentation *Proc. IEEE Conf. Comput. Vis. Pattern Recog* 580–87.

[11] Szegedy, A. Toshev, and D. Erhan 2013 Deep neural networks for object detection *Proc. Adv. Neural Inf. Process. Syst* 2553–61.

[12] J. H. Hosang, R. Benenson, P. Doll´ar, and B. Schiele 2015 What makes for effective detection proposals? *arXiv preprint arXiv:1502.05082.*

[13] Y.Jia,E.Shelhamer,J.Donahue,S.Karayev,J.Long,R.Girshick, S. Guadarrama, and T. Darrell. Caffe 2014 Convolutional architecturefor fast feature embedding *Proc. of the ACM International Conf. on Multimedia.*

[14] A. Krizhevsky, I. Sutskever, and G. Hinton 2012 ImageNet classification with deep convolutionalneural networks *InNIPS.*

[15] S. Lazebnik, C. Schmid, and J. Ponce 2006 Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR.*

[16] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel 1989 Backpropagation applied to handwritten zip code recognition *Neural Comp*

[17] M. Lin, Q. Chen, and S. Yan 2014 Network in network. In *ICLR.*

[18] T. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, P. Doll´ar, and C. L. Zitnick 2014 Microsoft COCO: common objects in context. *arXiv e-prints, arXiv:1405.0312 [cs.CV].*