A Review of Object Detection Models based on Convolutional Neural Network

F. Sultana  
Department of Computer Science  
University of Gour Banga  
West Bengal, India  
Email: sfarhana@ieee.org

A. Sufian  
Department of Computer Science  
University of Gour Banga  
West Bengal, India  
Email: sufian@ieee.org

P. Dutta  
Department of CSS  
Visva-Bharati University  
West Bengal, India  
Email: paramartha.dutta@gmail.com

Abstract

Convolutional Neural Network (CNN) has become the state-of-the-art for object detection task. In this paper, we have explained different object detection models based on CNN. We have categorized those detection models according to two different approaches: two-stage approach and one-stage approach. Through this paper, we have shown advancements in object detection model from R-CNN to latest RefineDet. We have discussed the model description and training details of each model. We have also drawn a comparison among those models.

Keywords

Convolutional Neural Network, Deep Learning, Mask R-CNN, Object Detection, R-CNN, RetinaNet, Review, YOLO

I. INTRODUCTION

In computer vision, the problem of estimating the class and location of objects contained within an image is known as object detection problem. Around a decade before 2013, object detection was done by using hand crafted machine learning features like shift invariant feature transform (SHIFT) [1] and histogram of oriented gradients (HOG) [2]. The best performing models among those gains around 35% mean average precision (mAP) on PASCAL VOC Challenge dataset [3]. But in 2010-2012 the progress in object detection has been slowed down. The best performing models gained their result building complex ensemble systems and applying little changes in successful models. Also the representation of SHIFT and HOG could only associate with the complex cells in V1 of Primary Visual Cortex. But both the structure of visual cortex of animals and the object recognition process in computer vision suggest that there might be hierarchical, multi-level processes for computing features that may contain more information for visual recognition. Inspired by this idea Kunihiko Fukushima built “neocognitron” [4] which is a hierarchical, shift-invariant multi-layered neural network for visual pattern recognition. In 1990, LeCun et al. extended neocognitron and introduced the CNN “LeNet-5” [5][6] which can be trained using supervised learning through stochastic gradient descent (SGD) [7] via backpropagation (BP) [8]. After that, the progress of CNN was stagnant for few years. In 2012, the success of “AlexNet” [9] inspired researchers of different field of computer vision such as image classification, localization, segmentation, object detection etc. and we have got many different classification models [10], object detection models like R-CNN [11], SPP-net [12], Fast R-CNN [13], YOLO [14], RetinaNet [15] etc. in successive years.

In this study, we have tried to give a review of different object detection models based on CNN. We have described architecture and training details of different object detection models based on CNN in section 2 with some subsections. In section 3 we have drawn a comparison between various object detection models. Finally, we have concluded our paper in section 4.

II. ARCHITECTURE OF DIFFERENT OBJECT DETECTION MODELS

The CNN based object detection models can be categorized into two different categories: (i) two stage approach and (ii) one stage approach.

A. Two Stage approach:

In the two stage approach as shown in figure 1, a set of sparsely connected region proposals is generated first and then these proposals are fed into CNN based module for further classification and bounding box regression.
Fig. 1: Two stage approach

1) R-CNN: R-CNN [13] is the first CNN based object detection model. The architecture of R-CNN contains three different blocks as shown in figure 2. In the first block, the authors have used selective search [16] algorithm to generate around 2000 class-independent region proposals from each input image. In the second block, following the architecture of AlexNet [9] they have used a CNN with five convolutional (conv) layers and two fully connected (FC) layers to extract fixed length feature vector from each region proposal. As CNN requires fixed sized image as input, the authors have used affine image warping [17] to get fixed sized input image from each region proposals regardless of their size or aspect ratio. Then those warped images are fed into individual CNN to extract fixed length feature vectors from each region proposal. The third block classifies each region proposal with category specific linear support vector machine (SVM) [18].

   a) Training details: The authors have pre-trained the CNN of their model using ILSVRC-2012 [19] [20] dataset. Then they have changed the ImageNet specific 1000-way softmax classifier with a 21-way classifier (for the 20 VOC classes and one background class) and trained the CNN parameters using SGD with warped region proposals taken from PASCAL VOC [3] images. If the IoU overlap of region proposals with corresponding ground truth box is \( \geq 0.5 \), they treat it as positive for that box class and rest as negative. The learning rate of SGD was 0.001. In each SGD iteration they uniformly sampled 32 positive foreground and 96 negative background window to construct a mini batch of size 128. After feature extraction and application of training labels, they optimize one linear SVM class using hard negative mining [21] method since the training data was too large to fit in a memory.

   R-CNN used conventional CNN models to classify region proposals. CNNs need fixed sized input image but region proposals are arbitrary in size. So the scale, aspect ratio and originality of an image are got compromised due to cropping, warping, predefined scales etc.

2) SPP-net: In a CNN, conv layers actually do not need fixed size image as input. But FC layers require fixed length vectors as input. On the other hand, spatial pyramid pooling (SPP) [22][23] can generate fixed sized output regardless of input size. So, He et al. includes SPP layer in between conv layers and FC layers of R-CNN and introduced SPP-net [12] as shown in figure 3. SPP layers pool the features and generates fixed length output from variable length region proposals, which are fed into FC layers or other classifiers. In this way, SPP-net made it possible to train and test the model with image of varying sizes and scales. Thus it increases scale invariance and reduces overfitting.

   a) Training details: The authors have followed [11][24] to train their SVM classifier. They have used ground truth box to generate positive samples. If the IoU of ground truth box and positive window is \( \leq 0.3 \) then they considered that window as negative sample. If negative sample - negative sample overlap is \( \geq 0.7 \), then that sample is removed. They have also applied standard hard negative mining [25] to train the SVM. At test time, the classifier is used to score the candidate windows. Then the authors used non-maximum suppression [25] (threshold of 0.3) on the scored windows. They followed [11] to fine tune only the FC layers of their model. In each mini-batch, the ratio of positive and negative samples was 1:3. They started training 250k mini-batches with the learning rate 0.0001, and then 50k mini-batches with 0.00001. Also following [11] they used bounding box regression on the pooled features from fifth convolutional layer. If the IoU of a window and a ground truth box is \( > 0.5 \) then that is used in bounding box regression training.

   As SPP-net uses CNN once on the entire image and region proposals are extracted from last conv feature map, the network is much faster than R-CNN.

3) Fast R-CNN: R-CNN and SPP-net had some common problems: (i) multi-stage pipeline training (feature extraction, network fine tuning, SVM training, bounding box regression), (ii) expensive training in terms of space and time and also (iii)
slow object detection. Fast R-CNN [13] tries to overcome all of the above limitations. Ross Girshick modified R-CNN and proposed Fast R-CNN which is a single-stage training algorithm that learns to classify region proposals and corrects their spatial locations together. Faster R-CNN can train very deep detection network like VGG-16 [26] 9× faster than R-CNN and 3× faster than SPP-net.

The conv layers of Fast R-CNN takes entire image and a set of object proposals as input instead of using a CNN for each region proposals as R-CNN. From the produced feature map, region of interests (RoI) are identified using selective search method. Using a RoI pooling layer the authors reshaped them into a fixed length feature vector. Then FC layers take feature vectors as input and passed the output to two sibling branch in output layer for classification and bounding box regression. Figure 4 demonstrates the architecture of Fast R-CNN.

a) Training details: The authors have experimented using three pre-trained ImageNet network with 5 maxpooling layers in between five and thirteen conv layers. During initialization of those pre-trained network, the authors first replaced the last maxpooling layer with a RoI pooling layer. Then last fully connected layer and softmax layer of the networks are replaced with two sibling output layers for classification and bounding box regression. The networks are also modified to take two inputs: a list of input images and a list of RoIs present in those images. The authors have trained Fast R-CNN using stochastic gradient descent (SGD) with hierarchically sampled mini-batches. They have used multi-task loss to jointly train their network for classification and bounding box regression. They have followed [13] and [12] to chose the ratio of positive and negative RoIs for training their model.

4) Faster R-CNN: R-CNN, SPP-net and Fast R-CNN depend on the region proposal algorithm. All of these methods exposed that computation of region proposals is slow and time consuming and it affects the performance of the network. Ren et al. proposed Faster R-CNN [27], where they replaced previous region proposals method with region proposal network (RPN). An RPN is a fully convolutional network (FCN) [28] that takes an image of arbitrary size as input and outputs a set of rectangular candidate object proposals, each with an objectness scores to detect an object or not in the box.

Similar to Fast R-CNN, the entire image is provided as an input to the conv layers of Faster R-CNN to produce convolutional feature map. Then instead of using selective search algorithm on the feature map to identify the region proposals, a RPN is used to predict the region proposals. The predicted region proposals are then reshaped using a RoI pooling layer and fed into FC layers to classify the image within the proposed region and predict the offset values for the bounding boxes. Multiple region proposals are predicted by the FC layers. The region proposals detected by the sliding window are called anchors. When the anchor boxes are detected, they are selected by applying a threshold over the “objectness” score to keep only the relevant boxes. These anchor boxes and the feature maps, computed by the initial CNN model, feeds a Fast R-CNN model.

a) Training details: The RPN uses a pre-trained model over the ImageNet dataset for classification and it is fine-tuned on the PASCAL VOC dataset. Then the generated region proposals with anchor boxes are used to train the Fast R-CNN. To train RPN, each anchor is assigned to a binary class label. Positive label is assigned to two kinds of anchors: (i) the anchor/anchors with the highest Intersection-over-Union (IoU) overlap with a ground-truth box, or (ii) an anchor that has an IoU overlap > 0.7 with any ground-truth box. A negative label is assigned to an anchor if its IoU ratio is < 0.3 for all ground-truth boxes. They simply followed the multi-task loss of Fast R-CNN to train their network. They train their RPN with back propagation and SGD and also followed the ‘image centric’ sampling strategy of Fast R-CNN.

5) Mask R-CNN: Mask R-CNN [29] extends previous object detection techniques to go one step further and locate exact pixels of each object (instance segmentation [30][31]) instead of just bounding boxes. Mask R-CNN has the exact region proposal network(RPN) from Faster R-CNN to produce region proposals. Then the authors applied RoIAlign layer on the region proposals instead of RoI pooling layer to align the extracted features with the input location of an object. The aligned RoIs are then fed into the last section of the Mask R-CNN to generate three output: a class label, a bounding box offset and a binary object mask. The architecture of Mask R-CNN is shown in figure 6.

a) Training details: The authors have set all hyperparameters of their model following [13][27]. An RoI is considered as positive if it has IoU with a ground-truth box of at least 0.5 and negative otherwise. The multi-task loss function of Mask R-CNN combines the loss of classification, localization and segmentation mask. The mask loss is defined only on positive RoIs. The mask target is the intersection between an RoI and its associated ground-truth mask.
B. One stage approach

In the one stage approach, classification and regression is done in a single shot using regular and dense sampling with respect to locations, scales and aspect ratio.

1) YOLO: The You Only Look Once (YOLO) [14] is a single stage network model which predicts class probabilities and bounding boxes directly from input image using a simple CNN. The model divides the input image into fixed number of grids. Each cell of this grid predicts fixed number of bounding boxes with a confidence score. The confidence score is calculated by multiplying the probability to detect the object with the IoU between the predicted and the ground truth boxes. The bounding boxes having the class probability above a threshold value is selected and used to locate the object within the image.

   a) Training details: YOLO has 24 conv layers and two FC layers. They have increased the resolution of input image and used batch normalization [33] to bounding box coordinates, height and width to increase accuracy. The final layer of the network used linear activation function and rest of the layers used leaky rectified linear activation [34]. Uses of sum squared error in the output layer made their model to be optimized easily. Though YOLO predicts multiple bounding boxes per grid cell, the authors have chosen one object prediction per bounding box during training. These strategy leads to better prediction with respect to sizes, aspect ratio, classes and overall recall. They have trained/validated/tested their network on PASCAL VOC 2007 and 2012 datasets. For training they have used mini-batch size of 64, momentum of 0.9 and weight decay of 0.005. Their initial learning rate was 0.001, then gradually increased it to 0.01 for 70 epochs, then decreased it by a factor of 10 for each 30 epochs. To avoid overfitting they have used dropout [35] and heavy data augmentation.

2) SSD: The single shot multibox detection (SSD) [32] model takes an image as input which passes through multiple conv layers with different sizes of filters (10x10, 5x5 and 3x3). Feature maps from conv layers at different position of the network are used to predict the bounding boxes. They are processed by a specific conv layers with 3x3 filters, called extra feature layers, to produce a set of bounding boxes. Similar to the anchor boxes (default box) of the Fast R-CNN, anchor box of SSD has 4 parameters: the coordinates of the center, the width and the height. At the time of predicting bounding boxes, the model produces a vector of probabilities corresponding to the confidence over each class of object. In order to handle the scale, SSD predicts bounding boxes after multiple conv layers. Since each conv layer operates at a different scale, it is able to detect objects of various scales. This detector achieves a good balance between speed and accuracy.

   a) Training details: During training, for each ground truth box, the authors have selected those anchor boxes that vary with respect to location, aspect ratio and scales. Then they have matched anchor with ground truth box with the best jaccard overlap [36]. They have selected those anchor boxes whose jaccard overlap to the ground truth is > 0.5. To handle different object scales they have used different feature maps from different layers in their network. They have designed the anchors in such a way that certain feature maps learn to be responsive to particular scales of the objects. They have trained their model using hard negative mining with negative and positive default box ratio of 3:1 and using heavy data augmentation.

3) YOLO9000: YOLO9000 [37] is a real-time object detector capable of detecting more than 9000 object categories by jointly optimizing detection and classification. The authors have used WordTree [38] hierarchy to combine data from various sources.
such as ImageNet and MS COCO [39] dataset and proposed an algorithm to jointly train the network for optimization on those different dataset simultaneously. They also provide improvements over the initial YOLO model to enhance its performances without decreasing its speed.

a) Training details: The authors first produced YOLOv2 [40] improving the basic YOLO system. Then they have trained their model using joint training algorithm on more than 9000 classes from ImageNet and from MS COCO detection data. They have used batch normalization instead of dropout and their input image size was $416 \times 416$ to capture center part of an image. They have removed a pooling layer from the network to capture high resolution features. Anchor boxes are used for predicting bounding boxes. The dimensions of anchor box are chosen using k-means clustering on the training set. They have used fine grained features to detect smaller object and also used multi-scale training by choosing new image size randomly per 10 batches to make the model scale invariant.

4) RetinaNet: RetinaNet [15] is a simple one-stage, unified object detector which works on dense sampling of object locations in an input image. As shown in the figure 9, the model consists of a backbone network and two task-specific subnetworks. As the backbone network, they used feature pyramid network (FPN) [41]. On the backbones output, the first subnetwork performs convolutional object classification and the second subnetwork performs convolutional bounding box regression. These subnetworks are basically small fully convolutional network (FCN) [42] attached to each FPN level for parameter sharing.

a) Training details: The authors have proposed a loss function called Focal loss to handle one stage object detection scenario where the network suffers from extreme foreground and background class imbalance during training. They have used this loss function on the output of the classification subnet. RetinaNet is trained with synchronized SGD over 8 GPUs with mini-batch of size 16. The model is trained for 90k iterations with an initial learning rate of 0.01, which is then decreased by 1/10 at 60k and again at 80k iterations. A weight decay of 0.0001 and a momentum of 0.9 are used. The training loss of their model is the sum of the focal loss and the standard smooth L1 loss used for box regression.

5) RefineDet: RefineDet [43], as shown in figure 10 is a single-shot object detector based on a feed forward convolutional network. This network consists of two interconnected modules (i) the anchor refinement module (ARM) (ii) the object detection module (ODM). The ARM module reduces search space for the classifier by filtering negative anchors and roughly adjusts the sizes and locations of the anchors for providing better initialization for the successive regressors. Taking the refined anchors from the previous module as input the ODM module improves the regression accuracy and predict multiple class labels. The transfer connected block (TCB) transfers the features in the ARM module to predict locations, sizes and class labels of object in the ODM module.

a) Training details: The authors have used VGG-16 and ResNet 101 [44] network. These networks are pre-trained on ImageNet classification and detection dataset. Instead of sixth and seventh FC layers of VGG-16, they have used conv layers. Also to capture high level information and detecting objects in multiple scales, they have added two conv layers at the end of the VGG-16 network and one residual block at the end of the ResNet-101. The parameters of extra two layers of VGG-16 are initialized using “xavier” [45] method. To deal with different object scales, they have selected 4 feature layers with stride size 8, 16, 32 and 64 pixels for both CNN networks. Then they have associated those strides with several different anchors for prediction. The authors determined the matching of anchor and ground truth box using jaccard overlap following [32]. Also they have trained their model using hard negative mining following [32]. They set the default training batch size to 32 and then fine-tuned the network using SGD with 0.9 momentum and 0.0005 weight decay.

III. COMPARATIVE RESULT

In table I, we have shown comparative performance of different object detection models on PASCAL VOC and MS COCO dataset. From second to tenth columns of the table representing the mean average precision (mAP) in measuring accuracy and last two columns are giving a overview of those models speed and organizational approach.
TABLE I: Comparison of different object detection models

| Model     | PASCAL VOC 2007 | PASCAL VOC 2010 | PASCAL VOC 2010 | COCO 2015 (IoU =0.5) | COCO 2015 (IoU =0.75) | COCO 2015 (Official Metric) | COCO 2016 (IoU =0.5) | COCO 2016 (IoU =0.75) | COCO 2016 (Official Metric) | Real time speed | Stage |
|-----------|-----------------|-----------------|-----------------|----------------------|-----------------------|-----------------------------|----------------------|----------------------|-----------------------------|---------------|-------|
| R-CNN     | 62.4%           |                 |                 |                      |                       |                             |                      |                      |                             | No            | Two   |
| SPP-net   |                 |                 |                 |                      |                       |                             |                      |                      |                             | No            | Two   |
| Faster R-CNN | 70.0%       | 68.8%           | 68.4%           |                      |                       |                             |                      |                      |                             | No            | Two   |
| Faster R-CNN | 78.8%       | 75.9%           |                 |                      |                       |                             |                      |                      |                             | No            | Two   |
| Mask R-CNN |                 |                 |                 | 62.3%                | 43.3%                 | 39.8%                       |                      |                      |                             | No            | Two   |
| YOLO      | 63.7%           | 57.9%           |                 |                      |                       |                             |                      |                      |                             | Yes           | One   |
| SSD       | 83.2%           | 82.2%           | 48.5%           | 30.3%                | 31.5%                 |                             |                      |                      |                             | No            | One   |
| YOLO9000  | 78.6%           |                 |                 | 44.0%                | 19.2%                 | 21.6%                       |                      |                      |                             | Yes           | One   |
| RetinaNet |                 |                 | 37.8%           |                      |                       |                             |                      |                      |                             | Yes           | One   |
| RefineDet | 85.6%           | 86.8%           | 41.8%           |                      |                       |                             |                      |                      |                             | Yes           | One   |

IV. CONCLUSION

In this study, we have discussed the advancements of different object detection models based on CNN. We have also shown that those models can be categorized into two different approaches: two-stage and one-stage. Two-stage models gave higher accuracy than one-stage models in object detection but they are slower. R-CNN, SPP-net and Fast R-CNN were slow because of external region proposal Network. Faster R-CNN overcame that problem using RPN. Mask R-CNN adds instance segmentation to the architecture of previous model. YOLO, SSD gave us a way for fast and robust object detection. RetinaNet focuses on improving loss function for better detection. RefineDet combined the merit of both two-stage and one-stage approach and it has achieved state-of-the-art performance. Through this paper, we have shown that the progress of various models are mainly because of better CNN models, new detection architecture, different pooling method, novel loss design etc. The improvements of different models give us the hope for more accurate and faster real time object detection.

REFERENCES

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” Int. J. Comput. Vision, vol. 60, no. 2, pp. 91–110, Nov. 2004. [Online]. Available: https://doi.org/10.1023/B:VISI.0000029664.99615.94
[2] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), vol. 1, June 2005, pp. 886–893 vol. 1.
[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, Jun 2010. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4
[4] K. Fukushima, “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,” Biological Cybernetics, vol. 36, no. 4, pp. 193–202, Apr 1980. [Online]. Available: https://doi.org/10.1007/BF00344251
[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” Neural Comput., vol. 1, no. 4, pp. 541–551, Dec. 1989. [Online]. Available: http://dx.doi.org/10.1162/neco.1989.1.4.541
[6] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” in Proceedings of the IEEE, 1998, pp. 2278–2324.
[7] L. Bottou, “Large-scale machine learning with stochastic gradient descent,” in Proceedings of COMPSTAT’2010. Heidelberg: Physica-Verlag HD, 2010, pp. 177–186.
[8] Y. L. Cun, “A theoretical framework for back-propagation,” 1988.
[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems 25, 2012, pp. 1097–1105.
[10] F. Sultana, A. Safian, and P. Dutta, “Advancements in image classification using convolutional neural network,” in ICRICIN, 2018.
[11] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, 2014.
[12] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” CoRR, vol. abs/1406.4729, 2014. [Online]. Available: http://arxiv.org/abs/1406.4729
[13] R. B. Girshick, “Fast R-CNN,” CoRR, vol. abs/1504.08083, 2015. [Online]. Available: http://arxiv.org/abs/1504.08083
[14] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” CoRR, vol. abs/1506.02640, 2015. [Online]. Available: http://arxiv.org/abs/1506.02640
[15] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” CoRR, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002
[16] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, “Selective search for object recognition,” International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, Jun 2010. [Online]. Available: https://doi.org/10.1007/s11263-009-0275-4
[17] G. Wolberg, Digital Image Warping, 1st ed. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994.
[18] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support vector machines,” IEEE Intelligent Systems and their Applications, vol. 11, no. 4, pp. 18–28, July 1998.
[19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” in CVPR09, 2009.
