International Journal of Advance Research in Science and Engineering
Volume No. 10, Issue No. 12, December 2021
www.ijarse.com
IJARSE
ISSN 2319 - 8354

A COMPREHENSIVE REVIEW ON OBJECT DETECTION ALGORITHMS

Dr.B.Gnana Priya

Assistant Professor,Department of Computer Science and Engineering,
Faculty of Engineering and Technology, Annamalai University

ABSTRACT

Object detection a part of computer vision is one of the most interesting and challenging field. It has been applied widely in various tasks such as activity recognition, face detection, traffic monitoring, surveillance, autonomous driving, image annotation and so on. The purpose of object detection is to locate the instances of semantic objects of a certain class. Traditional object detection methods are built on handcrafted features and shallow trainable architectures. Their performance easily stagnates by constructing complex ensembles which combine multiple low-level image features with high-level context from object detectors and scene classifiers. With the rapid development of deep learning networks for detection tasks, the performance of object detectors has been greatly improved. With the introduction of more powerful tools, they are able to learn semantic, high-level and deeper features accurately. These models behave differently in network architecture, training strategy and optimization function, etc. In this paper, we provide a review on latest improvements in object detection frameworks.

Keywords: R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, YOLO, SSD

1. INTRODUCTION

Object detection has gained a lot of attention and is a fast emerging field in recent years due to its need in wide range of applications and recent technological breakthroughs. It has a wide range of applications including, monitoring security, autonomous driving, transportation surveillance, drone scene analysis, and robotic vision. Due to vast developments in deep learning models, there is a fast evolution of object detection techniques. Deep learning models have been widely used in computer vision tasks for general object detection and domain-specific object detection. Many of the state-of-the-art objects detection algorithms uses deep learning networks as their backbone. Deep networks are used to extract features from input images or videos followed by operations such as classification or localization with respect to the problem. Some of the domains of object detection include multi category object detection, face detection, activity recognition, edge detection, pose detection, etc.

In Deep Neural Networks a more significant gain is obtained with the introduction of Region based CNN detectors (R-CNN) [1]. Since the proposal of R-CNN, a series of significant contributions have been made which promote the development of general object detection by a large margin. A number of improved models have been suggested, including Fast R-CNN which jointly optimizes classification and bounding box regression tasks [2],

Faster R-CNN which takes an additional sub network to generate region proposals [4] and YOLO which accomplishes object detection via a fixed-grid regression [3]. All of them bring different degrees of detection performance improvements over the primary R-CNN and make real-time and accurate object detection become more achievable.

Image object detection algorithms are broadly classified into two major types. One stage detectors like YOLO and SSD [5]. Two stage detectors such as Faster R-CNN. There are several advantages with both the types. The one-stage detectors directly propose predicted boxes from input images without any region proposal step. So, they are efficient in time and can be used for real-time tasks. In two stages detector the major processing is divided into two steps. The two stages can be divided by Region of Interest pooling layers. As an example, in Faster R-CNN the first stage is a RPN (Region Proposal Network) which proposes candidate object bounding boxes. In the second stage features are extracted by RoIPool operation from each candidate box for any classification and bounding-box regression tasks [6]. Two-stage detectors have high localization and object recognition accuracy, whereas the one-stage detectors achieve high inference speed.

2. R-CNN

R-CNN (Region CNN) is widely used for multiple objects detection present in an image. R-CNN works in four steps: i) Region proposal generation ii) CNN based feature extraction iii) Classification or Localization and iv) Bounding box regressor . It is based on the principle that a single object will dominate in a given region of an image. Regions in an image can be divided based on colors, texture, scales, etc. About 2000 Region proposals are generated using selective search algorithm. The algorithm generates a small set of high-quality object locations using the combination of segmentation and exhaustive search method. The image segmentation aims to generate object locations based on the structure of the image, while the exhaustive search is responsible to find all possible object locations. The algorithm uses a bottom-up approach for grouping of the image regions to generate a hierarchy of small to large regions.

The selective search algorithm works as follows: Initially, the regions are sub-segmented based on the criteria that each region belongs to at most one object. Next, Greedy algorithm is used to recursively combine similar regions into larger ones. The generated regions are used to produce object locations. R-CNN uses region proposal along with CNN for multiple object detection. Region proposal is a set of candidate detection available to the detector. R-CNN selects a few windows rather than sliding windows over the entire image like CNN. The proposed regions are wrapped and are given as input to large CNN which acts as a feature extractor that extracts a fixed-length feature vector from each region. After passing through the CNN, R-CNN extracts a 4096-dimensional feature vector for each region proposal. Next, Support Vector Machine (SVM) used to extract features from CNN and classify the presence of the object in the region. The problems with R-CNN are: Classification of 2000 region proposals takes a large time to train the network. The selective search algorithm is not learning from the region proposals. The time taken for training and testing are quite large.

3. Fast R-CNN

Fast R-CNN, a faster version of R-CNN is a framework that uses deep ConvNets for object classification and object detection. Fast RCNN extracts features from an entire input image and then passes the region of interest (RoI) pooling layer to get the fixed size features. They are given as input to CNN for classification, followed by bounding box regression using fully connected layers. Unlike R-CNN, Fast R-CNN uses a single deep ConvNet to extract features for the entire image once. Here, instead of feeding the region proposals to the CNN, the input image is fed to the CNN to generate a convolutional feature map. Fast R-CNN uses a one stage training process to jointly train the network on each labeled RoI. Also, Fast R-CNN uses a RoI pooling layer to extract a fixed size feature map from region proposals of different size. ROI pooling layer is then fed into the fully connected layer for classification as well as localization. RoI pooling layer uses max pooling. It converts features inside any valid region of interest into a small feature map.

The improvement over R-CNN is that, instead of using 2000 ConvNets for each region of the image, a single deep Convolutional network that speeds processing is used. Fast R-CNN uses a single model for extracting features, generation of bounding boxes and classification unlike R-CNN that separate models for each one of the process. R-CNN uses SVM for classification while Fast R-CNN uses SoftMax for object classification. To increase detection accuracy, multitask loss is used to achieve an end to end training of Deep Convolutional Network. Like R-CNN, Fast R-CNN also uses selective search as a region proposal method to find the Regions of Interest(RoI). This is a slow and time consuming algorithm that is not suitable for large real time datasets.

4. Faster R-CNN

Faster R-CNN further improves the region based architecture. R-CNN and fast R-CNN uses selective search algorithm that takes 2 seconds for computation of each image. The proposal of RoI is slow and needs more running time. The Faster R-CNN uses another convolutional network called the Region proposal network (RPN) to generate the region proposal. The RPN efficiently predicts region proposals with a wide range of scales and aspect ratios. Region Proposal Network fastens the generation speed of region proposals, as it shares the full image convolutional features and a common set of convolutional layers with the detection network. This causes an overall improvement in feature representation. Here, for different sized object detection multi-scale anchors are used as reference. The anchors facilitate the process of generating various sized region proposals in a simple manner. There is no need of multiple scales of input images or features in this method. The region proposal is parameterized relative to a reference anchor box. Also, the distance between predicted box and its corresponding ground truth box is measured to optimize the location of the predicted box.

5. Mask R-CNN

Mask R-CNN is an extended work from Faster R-CNN developed mainly for instance segmentation in computer vision. Mask-RCNN outputs bounding boxes classes and mask, so that it is possible to separate all the objects in a subject. It works in two stages. In the first step, it generates region proposals in the location of each object given the input image. The second stage provides class prediction for object, generates mask and refines the bounding boxes. The two stages are linked using a backbone that consists of a bottom up and top down pathways. Generally VGG or ResNet is used for bottom up pathway to extract features from images. Top down pathway is used to generate feature pyramid map. They are both connected by convolution and adding operations between corresponding levels called lateral connections. Regardless of the adding parallel mask branch, Mask R-CNN can be seen a more accurate object detector. While Faster R-CNN has 2 outputs for each candidate object, a class label and a bounding-box offset, Mask R-CNN is the addition of a third branch that outputs the object mask. The additional mask output is distinct from the class and box outputs, requiring the extraction of a much finer spatial layout of an object. Faster RCNN with a ResNet backbone to extract features achieves excellent accuracy and processing speed.

6. YOLO

YOLO (You Only Look Once), a single stage object detector uses a single convolutional neural network to predict multiple bounding boxes and class probabilities simultaneously. YOLO is immensely fast and accurate because it trains on full images and directly optimizes detection performance. The power of YOLO object detection algorithm is that, it finds all objects in an image grid simultaneously. Unlike region based techniques, YOLO uses the entire image during training and testing and learns the contextual information about classes with an understanding of their appearances. The network contains 24 convolutional layers and 2 fully connected layers. Some of the convolutional layers construct ensembles of inception modules with 1 × 1 reduction layers followed by 3 × 3 convolutional layers. This algorithm takes one image at a time and split it into an S x S grid. Each grid cell predicts only one object. If the center of an object falls into a grid cell then that grid cell is responsible for detecting that object. On each grid image classification and localization is applied. It outputs confidence score, all bounding boxes and the class probabilities for these boxes. It uses concepts like Intersection over union (IOU) and non-max suppression. Improved versions of YOLO such as YOLOv2, YOLOv3, YOLOv4, YOLOv5 and PP-YOLO were released recently.

7. SSD

Liu et al. proposed Single Shot MultiBox Detector (SSD) which uses a set of default anchor boxes with different aspect ratios and scales to discretize the output space of bounding boxes. The difficulty in dealing with small objects by YOLO is rectified here. The small object prediction struggle is caused by spatial constraints imposed on bounding box predictions. Also, due to multiple downsampling operations, YOLO find difficulty in generalizing objects with different aspect ratios and configuration. So, it produces relatively coarse features. SSD has a number of anchor boxes, each with different aspect ratios and scales to discretize the output space, instead of

using fixed grids for each feature map as in YOLO. The multiple feature maps with various resolutions are fused for predicting objects with various sizes. With VGG16 as backbone architecture, several convolutional layers are added to the network for predicting the default anchor boxes and their associated confidences. SSD uses 8732 boxes. This helps with finding the default box that most overlaps with the ground truth bounding box containing objects. The network is trained with a weighted sum of localization loss and confidence loss. Final detection results are obtained by conducting NMS on multi-scale refined bounding boxes. SSD outperforms the Faster R-CNN on PASCAL VOC and COCO dataset in terms of accuracy.

8. Conclusions

A set of related tasks for identifying objects in image and video is known as object detection. Deep learning based object detection methods are gaining importance nowadays due to its quick ability in learning. Also, they are preferred for their ability in dealing with occlusion, scale transformation and background switches. This paper provides a detailed review on deep learning based object detection frameworks. Region-Based Convolutional Neural Networks are a family of techniques for addressing object localization and recognition tasks. YOLO is a family of techniques for object recognition designed for speed and real-time use, while SSD is designed for object detection in real-time.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587, June 2014.

[2] R. Girshick, "Fast r-cnn," in 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440–1448, Dec 2015.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in CVPR, 2016.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards realtime object detection with region proposal networks," in NIPS, 2015, pp. 91–99.

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in Computer Vision – ECCV 2016 (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), (Cham), pp. 21–37, Springer International Publishing, 2016.

[6] K. He, G. Gkioxari, P. Dollr, and R. Girshick, "Mask r-cnn," in 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980– 2988, Oct 2017.

[7] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525, July 2017.

[8] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," CoRR, vol. abs/1804.02767, 2018.

[9] Q. Zhao, T. Sheng, Y. Wang, Z. Tang, Y. Chen, L. Cai, and H. Ling, "M2det: A single-shot object detector based on multi-level feature pyramid network," arXiv preprint arXiv:1811.04533, 2018.

[10] K. Chen, J. Li, W. Lin, J. See, J. Wang, L. Duan, Z. Chen, C. He, and J. Zou, "Towards accurate one-stage object detection with ap-loss," arXiv preprint arXiv:1904.06373, 2019.

[11] R. Dong, D. Xu, J. Zhao, L. Jiao, and J. An, "Sig-nms-based faster r-cnn combining transfer learning for small target detection in vhr optical remote sensing imagery," IEEE Transactions on Geoscience and Remote Sensing, 2019

[12] M. Feng, H. Lu, and E. Ding, "Attentive feedback network for boundary-aware salient object detection," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1623– 1632, 2019.