딥러닝을 이용한 객체 검출 알고리즘

강동연
한양대학교 컴퓨터 공학과
e-mail:dongykang@gmail.com

Popular Object detection algorithms in deep learning

Dongyeon Kang
Dept of Computer Science, Han-yang University

Abstract

Object detection is applied in various field. Autonomous driving, surveillance, OCR(optical character recognition) and aerial image etc. We will look at the algorithms that are using to object detect. These algorithms are divided into two methods. The one is R-CNN algorithms [2], [5] ,[6] which based on region proposal. The other is YOLO [7] and SSD [8] which are one stage object detector based on regression/classification.

1. Introduction

In order to image understanding, we have to not only concentrate on distinction of different images, but also try to estimate the concepts and locations of objects in each image. This task is called as object detection which is the identification of an object in an image along with its localization and classification.

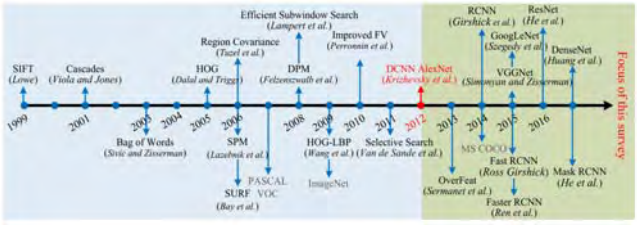

Figure 1. Milestones of object detection and recognition, including feature representations [1].

Object detectors have been making fast strides in accuracy and speed. Figure 1 shows that object detection has been gaining huge amounts of traction in recent years , when the first viable deep learning based object detector [2] was introduced. As one of the fundamental computer vision problems, the difference between object detection and classification is that in detection algorithms, Find a location within the image by drawing a bounding box around the object of interest. It is not necessary to have only one bounding box in the object detection case. There can be many bounding boxes within an image that represent different objects of interest, and some are not known in advance [3]. The main reason that you cannot solve this problem by building a standard convolution network and building a fully connected layer is that the length of the output layer is variable. This is because the number of occurrences of the objects of interest is not fixed. A naive approach to solve this problem would be to take different regions of interest from the image, and use a CNN to classify the presence of the object within that region. The problem with this approach is that the objects of interest might have different spatial locations within the image and different aspect ratios. Hence, you would have to select a huge number of regions [3] and this could computationally expensive. Therefore, algorithms like R-CNN [3], YOLO [7] and so on, have been developed to find these occurrences and make them fast.

There are currently two perspective of constructing object detectors-the region proposal based and regression/classification based. The former referred 1-stage object detector and the latter is called 2-stage object detector. In this paper, we shall start the review these popular algorithms in aspects of comparing these algorithms.

2. Object detector models

Region Proposal based

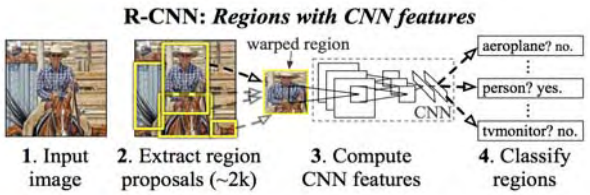

Figure 2. The flow chart of R-CNN [3]. Which consists of 3 stage: (1) extract bottom up region proposal. (2) computes features for each proposal using a CNN, and then (3) classifies each region with class-specific linear SVMs.

A. REGION CONVOLUTIONAL NETWORK (R-CNN)

In this model, to solve the problem of selecting a huge number of regions, proposed a method of extracting only 2k regions from an image [3] using selective search, which we call an region proposal.

The R-CNN adopts Selective Search [4] to generate about 2k region proposals for each image. The flow chart of R_CNN shows in Figure 2. Selective Search: 1) Generate initial sub-segmentation, we generate many candidate regions. 2) Use greedy algorithm to recursively combine similar regions into larger ones. 3) Use the generated regions to produce the final candidate region proposals. These 2k candidate region proposals are each warped or cropped into a fixed resolution and the convolutional neural network that produces a 4096-dimensional feature vector as output. The CNN acts as a feature extractor and the output dense layer consists of the features extracted from the image and the extracted features are fed into an SVM to classify the presence of the object within that candidate region proposal. Futhermore, to predicting the existence of an object within a regional proposal, the algorithm also predicts four values which are offset values to increase the precision of the bounding box. So, the offset values help in adjusting the bounding box of regional proposal. However, there are still some drawbacks of this model. It still takes a huge amount of time to train the network as you would have to classify 2k region proposals per image [3] and cannot be implemented in real time. Also, Selective Search algorithm is a fixed algorithm [4]. This could lead to generating of bad candidate regional proposals.

B. FAST R-CNN

Same author of R-CNN paper solves some drawbacks of R-CNN to make a faster object detection algorithm and it is called Fast R-CNN. The base of approach is similar to the R-CNN. The difference is that instead of feeding the region proposal to the CNN [5], feeding the input image to the CNN to generate the convolutional feature map. The R-CNN take every region proposal and runs them through the convolutional network [3]. Thus, everytime a region proposal is processed. So, Fast R-CNN aims to reduce this overhead by running the convolutional base just once. Then, a fixed-length feature vector is extracted from each region proposal with a region of interest (RoI) pooling layer. By using a RoI pooling layer, we reshape them into a fixed size so that it can be fed into a fully connected layer. From the RoI feature vector, we use a softmax layer to predict the class of the proposed region and also the offset values for the bounding box [5].

Due to Fast R-CNN don't have to feed 2k region proposals to the convolutional neural network every time, the convolution operation is done once per image and a feature map is generated from it. So, Fast R-CNN is faster than R-CNN.

C. FASTER R-CNN

Both of the above algorithms(R-CNN & Fast R-CNN) uses Selective Search to find out the region proposals. Selective Search was took a lot of time and slow process affecting the performance of the network. Faster R-CNN introduced the Region proposal network(RPN) [6] to replace Selective Search. Similar to Fast R-CNN, the Faster R-CNN sharing full image convolutional features with detection network. The architecture is presented in Figure 3.

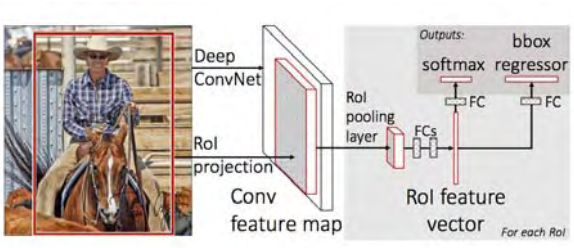

Figure 3. The architecture of Fast R-CNN [5].

RPN is achieved with a fully-convolutional network, which has the ability to predict object bounds and scores at each position simultaneously. Similar to Selective Search [4], RPN takes an image of arbitrary

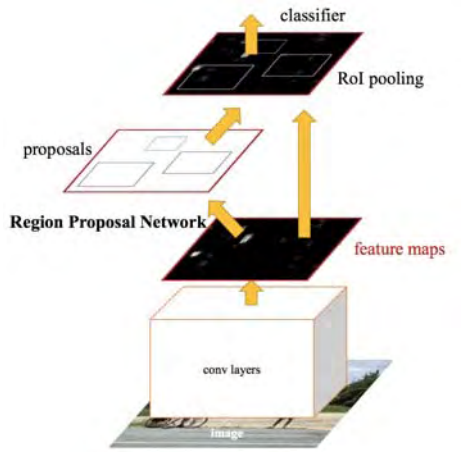

Figure 4. Overview of Faster R-CNN [6].

size to generate a set of rectangular object proposals. Figure 4 described the overview of Faster R-CNN. RPN operates on a specific convolutional layer with the preceding layers shared with object detection network. RPN is consists of two sub-networks. One is classification and the other is regression. The difference in two sub-networks is the shape of final feature map. The classification has w*h*(k*m)where w, h and k*m are the width, height and depth. The m represents the number of class. Feature map for regression is w*h*(k*4). The value of 4 is prediction for four offset coordinates for each anchors. Anchor is set of regions predefined shape and size. Thus, Faster R-CNN is faster and has end to end deep learning pipeline. Faster R-CNN improved state of the art accuracy with introduction of RPN which improved region proposal quality.

Regression/Classification based

D. YOU LOOK ONLY ONCE (YOLO)

All of the previous object detection algorithms use regions to localize the object within the image. In YOLO a single convolutional network predicts the bounding boxes and the class probabilities for these boxes. The basic idea of YOLO is exhibited in Figure 6. YOLO works that we take an image and split it into an S x S grid, within each of the grid we take m bounding boxes [7]. Each grid cell predicts B bounding boxes and their corresponding confidence score. The bounding box prediction has 5 components: (x, y, w, h, confidence). The (x, y) coordinates represent the center of the box, relative to the grid cell location. The (w, h)

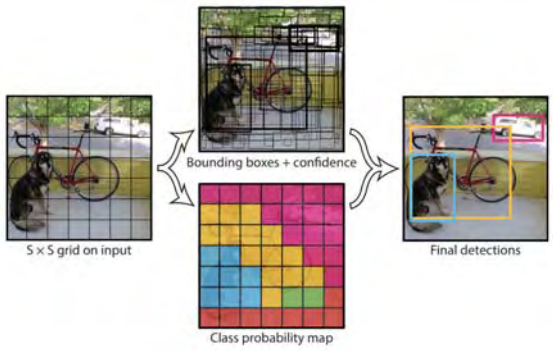

Figure 6. Main idea of YOLO [7].

box dimensions are also normalized to [0, 1], relative to the image size. YOLO consists of 24 convolution layers and 2 fully connected layers, of which some convolution layers construct ensembles of inception modules with 1 x 1 reduction layers followed by 3 x 3 convolution layers. YOLO is orders of magnitude faster(45 frames per second) [7] than other object detection algorithms. The limitation of YOLO algorithm is that it struggles with small objects within the image.

E. SINGLE SHOT MULTIBOX DETECTOR (SSD)

YOLO has difficulty in dealing with small objects in image, which is caused by strong spatial constraints imposed on bounding box predictions [5]. Given a specific feature map, instead of fixed grids adopted in YOLO, SSD takes advantage of a set of default anchor boxes with different aspect ratios and scales to discretize the output space of bounding boxes. To handle objects with various sizes, the network fuses predictions from multiple feature maps with different resolutions. The CNN in SSD is fully convolutional, which early layers are based on VGG network [9]. The architecture of SSD exhibited in Figure 7. Several convolutional layers progressively decreasing in size, and added to the end of the base network. The information in the last layer with low resolution may be too coarse spatially to allow precisely localization. SSD uses shallow layer with high resolution for detecting small objects [8]. Even SSD has more

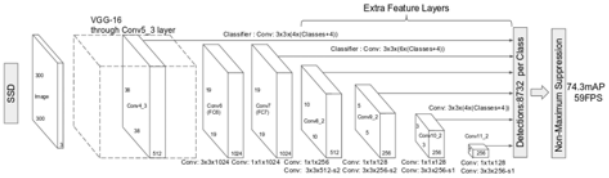

Figure 7. The architecture of SSD [8].

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	**74.8**	80.4	70.4
Faster [2]	73.2	76.5	79.0	70.9	**65.5**	**52.1**	83.1	84.7	86.4	52.0	**81.9**	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300	72.1	75.2	**79.8**	70.5	62.5	41.3	81.1	80.8	86.4	51.5	74.3	**72.3**	83.5	84.6	80.6	74.5	46.0	71.4	73.8	83.0	69.1
SSD500	**75.1**	**79.8**	79.5	**74.5**	63.4	51.9	**84.9**	**85.6**	**87.2**	**56.6**	80.1	70.0	**85.4**	**84.9**	**80.9**	**78.2**	**49.0**	**78.4**	72.4	**84.6**	**75.5**

Figure 8. PASCAL VOC2007 test detection results [8].

accurate and efficient than YOLO, it does not skilled at dealing with small objects.

3. Experimental results

The results of PASCAL VOC 2007 for the models using the algorithms we have seen so far are presented in Figure 9 [8]. 1-stage object detect models show relatively faster (SSD300 and SSD512 applies data augmentation for small objects to improve mAP). However, Figure 8 shows that SSD has relative difficulty in detecting small objects. Hence, Through the object detection algorithms [6],[7],[8] speed and accuracy are in a trade-off relationship [11]. In recent days, as the upgraded models are emerging, this trade-off relationship is gradually improving [10],[12].

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

Figure 9. Results on Pascal VOC2007 test [8].

3. Conclusion

This paper provides popular algorithms based on object detection methods which are divided into region proposal and regression/classification. Object detection models tend to infer localization and classification all at once to have an entirely differentiable network. Thus it can be trained from head to tail with backpropagation. Moreover, A trade-off [11] between high performance and realtime prediction capability is made between the recent models.

There are several more object detection architectures, which this paper has not touched upon. Especially when looking at real-time applications, YOLO-v2 [10] is often considered0 as an important architecture (fairly similar to SSD). Even though this was a review of representative object detection algorithms, we hope it gives you a basic understanding and a baseline for getting deeper knowledge.

References

[1] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, Matti Pietikainen, “Deep Learning for Generic Object Detection: A Survey” arXiv:1809.02165, 2018.

[2] P. Viola and M. Jones. “Rapid Object Detection using a Boosted Cascade of Simple Features“ in CVPR, 2001.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation” in CVPR, 2014.

[4] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective Search for Object Recognition” Int. J. of Comput. Vision, vol. 104, no. 2, pp. 154 - 171, 2013.

[5] R. Girshick, “Fast R-CNN” in ICCV, 2015.

[6] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks” in NIPS, 2015.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection” in CVPR, 2016.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot Multibox Detector” in ECCV, 2016.

[9] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition” arXiv:1409.1556, 2014.

[10] J. Redmon and A. Farhadi. “YOLO9000: Better, Faster, Stronger” In CVPR, 2017.

[11] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al. “Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors” in CVPR, 2017

[12] J. Redmon and A. Farhadi. “YOLOv3: An Incremental Improvement” arXiv:1804.02767, 2018.