A Review of Deep Convolutional Neural Networks For Object Detection (R-CNN and its variants)

A Written Creative Work submitted to the professor of
San Francisco State University
In partial fulfillment of
the requirements for
the course final project

by
Xiaoqian Yang
San Francisco, California
Fall Semester 2020

Abstract. This review begins with a brief introduction on the four fundamental visual recognition problems in the field of computer vision and compare traditional methods with deep learning based methods. Then, I mainly focus on object detection architectures: R-CNN, Fast R-CNN, Faster R-CNN and one instance segmentation architecture: Mask R-CNN. In each section, I introduce model workflow, then summarize the achievements and analyze the disadvantages. Finally, I compare these architectures and write down some of my thoughts.

1. Introduction

Image classification [1], object detection [2], semantic segmentation [3] and instance segmentation [4] are four fundamental visual recognition problems in the field of computer vision. The goal of image classification to assign an input image one label from a pre-defined category, its popular representative neural network architectures are Alexnet, VGG, Googlenet, Resnet, Densent, etc. Object detection not only recognized the object categories, but also predicts the location of each object instance via bounding boxes, its popular representative neural network architectures are R-CNN [2], Fast R-CNN [11], Faster R-CNN [12], YOLOV1-V4 [14], SSD [20], EfficientDet [21], etc. Semantic segmentation is the task of labeling each pixel of an image with a corresponding category, which provides an even richer understanding of an image [5]. And semantic segmentation's popular representative neural network architectures are FCN [22], Bisenet [23], Unet [24], etc. Instance segmentation combines object detection and semantic segmentation, aiming to identify different objects and assign each of the object instance of all classes a separate categorical pixel-level mask. And Mask R-CNN[13] is commonly used in this task. In this survey, I focus my attention on reviewing object detection. Currently, deep learning based object detection frameworks can be mainly divided into two families: (I) two-stage detectors, such as Region-based CNN (R-CNN) [2] and its variants [11, 12, 13] and (II) one-stage detectors, such as YOLO [14] and its variants [15, 16]. And I will especially focus on R-CNN and its variant: Fast R-CNN, Faster R-CNN and Mask R-CNN in this review.

2. Advantages of CNN against traditional methods

The pipeline of traditional object detection models can be divided into three stages: informative region selection, feature vector extraction and region classification. During the first stage, a multi-scale sliding window was used to scan the whole image, so it obviously generates too many redundant windows and it is computationally expensive. To extract visual features during the second stage, low-level visual descriptors such as SIFT (Scale Invariant Feature Transform) [6], HOG (Histogram of Gradients) [7], Harr [8] or SURF (Speed Up Robust Features) [9] are commonly used. However, it is not easy to manually design a robust feature descriptor to perfectly describe all categories of objects. During the third stage, region classifiers are used to assign categorical labels to the covered regions. Each stage of the detection pipeline was designed and optimized separately, therefore, recognition process has been very slow with small gains obtained by building ensemble systems and employing minor variants of successful methods.

But after the emerging of the Deep neural network architecture and Convolutional Neutral Networks, it has become more convenient and reliable to bridge the gaps which are present in the traditional object detection algorithms. Compare to traditional methods, convolutional neural networks have some advantages [10]. (I) It automatically detects the hierarchical feature without any human supervision. Hidden factors of input data can be disentangled through multi-level non-linear mappings. (II) Deeper architectures increased computation capability exponentially. (III) Make it possible to optimize several related tasks together. (IV) Benefitting from the large learning capacity of deep CNNs, some classical computer vision challenges can be recast as high-dimensional data transform problems and solved from a different viewpoint.

3. R-CNN

R-CNN which is short for "Region-based Convolutional Neural Networks" was proposed by Ross Girshick et al. in 2014 [2]. This paper presents a simple and scalable object detection algorithm that improves mAP (mean average precision) by more than 30% relative to the previous best result on PASCAL VOC 2012 – achieving a mAP of 53.3%.

3.1 R-CNN Model Workflow

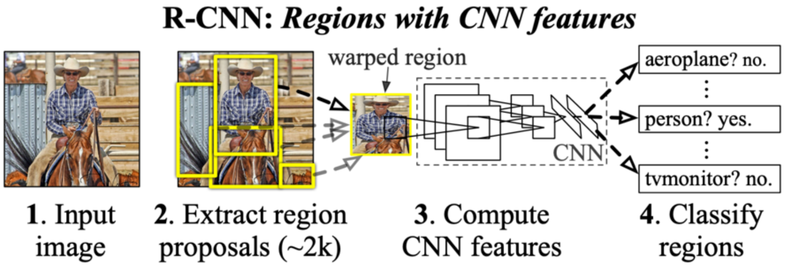


Figure 1: R-CNN Object Detection System Overview

First of all, we need to know that R-CNN consists of three models. A Convolutional Neutral Network for feature extraction, a set of linear SVM classifier for identifying objects, and a regression model for tightening the bounding boxes.

Step 1: Generating Region Proposals using the Selective Search Algorithm [17]

Take an input image, the first model generates approximately 2000 category-independent region proposals using selective search [17]. (step 1&2 in Figure 1).

Step 2: Formatting the proposed regions

Warp all pixels in a tight bounding box to the required size no matter what sizes or aspect ratios proposed regions have (step between 2&3 in Figure 1).

Step 3: Testing phase: Forward each region through Convolutional Neutral Network, CNN then extracts features for each region.

Training phase: Supervised pre-training & Domain-Specific fine-tuning

Pre-train the CNN on a large auxiliary dataset (ILSVRC 2012) using image-level annotations only with supervision. To adapt this pre-trained CNN to the detection task and the new domain (warped proposal regions), then use stochastic gradient descent (SGD) at a starting learning rate of 0.001 to continue fine-tuning train. During the training, all warped region proposals with 0.5 IoU (Intersection over Union) overlap with a ground-truth box as positives for that box's class and the rest will be treated as negatives. This process extracts a fixed-length feature vector from each proposal (step 3 in Figure 1).

Between these two types of training no changes are made to the network architecture, however, a lower learning rate for fine-tuning training is used so no valuable information is lost during the pre-training steps.

Step 4: Classify regions

Use a set of class-specific linear SVMs to classify/score the extracted feature vectors. (step 4 in Figure 1). Based on all region scores, a region will be rejected if it has an IoU overlap with a higher scoring selected region larger than a learn threshold, and this process is called Greedy Non-maximum suppression.

Step 5: Improve the bounding boxes

Run the region proposals through a simple linear regression, which will generate tighter bounding box coordinates once the object has been classified.

3.2 Achievements /Advantages of R-CNN

Object detection faces two challenges, one is that it requires localizing (likely many) objects within an image, the second one is that labeled data is scarce and the amount currently available is insufficient for training a large CNN. These two problems were addressed in this paper: R-CNN solves the first problem by operating within regions; and benefitting from supervised pre-training and domain-specific fine-tuning, it is able to train a high-capacity model with only a small number of labeled detection data.

R-CNN model is very efficient because of two properties: first, all CNN parameters are shared across all categories. Second, when compared to other common methods, the feature vectors computed by the CNN are low-dimensional.

The mAP is highly increased compare to the other methods at the same time without using CNN. See Figure 2.

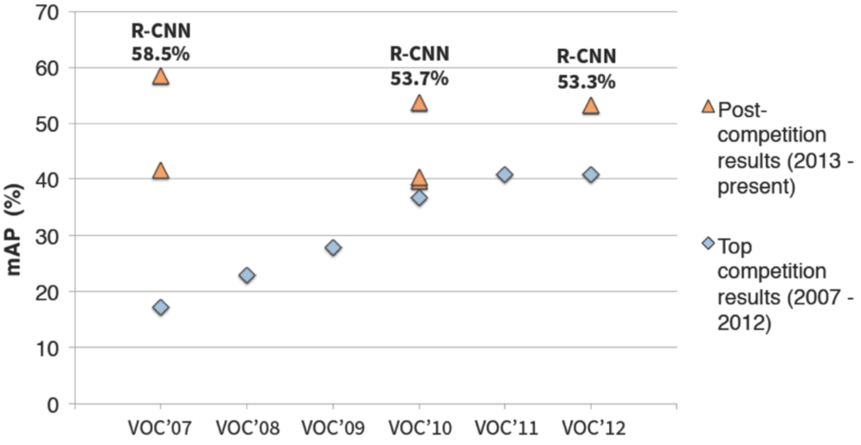


Figure 2. PASCAL VOC Challenge Dataset

R-CNN provides a rich hierarchy of image features, which provides an insight to what a network learns. Also, it bridges the gap between image classification and object detection by showing that a CNN can be generalized to common object detection tasks.

3.3 Disadvantages / Speed Bottleneck

Although R-CNN can be helpful for object detect, it still has some limitations and drawbacks.

First, it uses the Selective Search Algorithm to get the Regions of Interest (ROI) which is rigid and there is no learning process in this step, so sometimes bad region proposals are generated. Second, the process generates the CNN feature vector for every image region (N images * 2000) which takes a lot of time to train the network. Third, R-CNN consists of three models as we mentioned at the beginning of Section 3.1. And the process involves these three models separately without much shared computation. Fourth, lots of disk space is required to store the feature map of the region proposal.

4. Fast R-CNN

Fast R-CNN was proposed by Ross Girshick in 2015 [11]. It uses a single-stage training algorithm that improves the training procedure. Fast R-CNN unifies three independent models into one jointly trained framework and it increases training and testing speed along with the detection accuracy compared to R-CNN.

4.1 Fast R-CNN Model Workflow

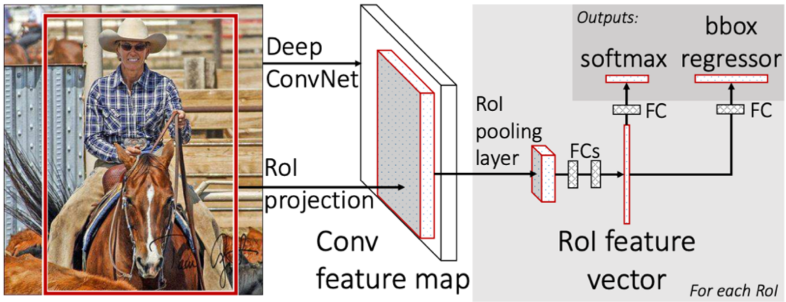


Figure 3: Fast R-CNN Architecture

Step 1: Take an image as input.

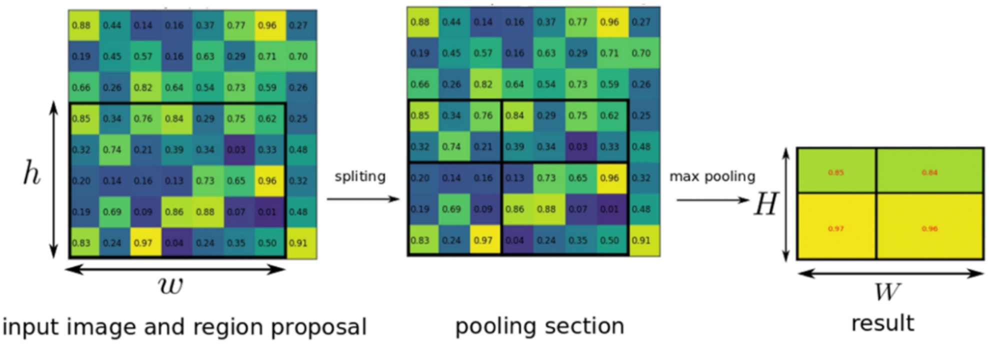
Step 2: Pass the image to a ConNet, which generates the regions of interests (RoIs) accordingly using selective search (approximately 2000 proposals per image).

Step 3: RoI pooling layer is then applied on each of the extracted RoIs to make sure all regions are of the same size.

Step 4: Pass these regions to a fully connected network to classify them, and it generates the bounding boxes using softmax and linear regression layers at the same time.

4.2 RoI(Region of Interest) Pooling and Multi-task Loss

It is time consuming to run CNN separately for each region proposal, so Fast R-CNN introduces a new technique – RoI Pooling. It can share the computation across the approximately 2000 proposals. The RoI pooling layer uses max pooling to convert the features of RoIs into a small feature map with a fixed-length feature vector, so that it can be passed into a sequence of fully connected layers.



0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

0.88	0.44	0.14	0.16	0.37	0.77	0.96	0.27
0.19	0.45	0.57	0.16	0.63	0.29	0.71	0.70
0.66	0.26	0.82	0.64	0.54	0.73	0.59	0.26
0.85	0.34	0.76	0.84	0.29	0.75	0.62	0.25
0.32	0.74	0.21	0.39	0.34	0.03	0.33	0.48
0.20	0.14	0.16	0.13	0.73	0.65	0.96	0.32
0.19	0.69	0.09	0.86	0.88	0.07	0.01	0.48
0.83	0.24	0.97	0.04	0.24	0.35	0.50	0.91

0.85	0.84
0.97	0.96

Figure 4: RoI Pooling Example

Figure 4 is an example of RoI Pooling. Let's say we have a region proposal (left) with h×w, and we want the output (right) to be the size of H×W after pooling. Then, the area for each pooling area (middle) is h/H × w/W. So with input RoI size of 5×7, and output of 2×2, the area for each pooling area is 2×3 or 3×3 after rounding. And output value is the maximum value within the pooling window for each grid, this is the same idea of conventional max pooling layer.

Other than the region proposal generation part, Fast R-CNN trains the whole model end-to-end at once to learn the category of objects and their bounding boxes' positions and sizes, therefore, the loss is multi-task loss.

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v).$$

In the loss function above, L_{cls} is the loss for true class u, L_{loc} is the loss for bounding box. u=0 is background class. [u≥1] means it is equal to 1 when u≥1

4.3 Achievements / Advantages of Fast R-CNN

Fast R-CNN combine all models into one network. In R-CNN, there are three models: CNN, SVM classifier and regression model. Fast R-CNN doesn't use SVM classifier anymore, instead it uses a softmax layer and a linear regression layer simultaneously on top of the CNN. The softmax layer outputs a classification and the regression layer outputs bounding box coordinates. In this way, Fast R-CNN compute everything in one single network.

Besides the achievements mentioned above, Fast R-CNN has other advantages: it has higher detection quality aka mean average precisions than R-CNN and SPPnet [19]; Training is a single-stage process which adopts a multi-task loss; Training can update all network layers; And no disk storage is required for feature storing.

Other findings of this paper's contribution: Not all convolutional layers should be fine-tuned; Single-scale processing balanced speed and accuracy; When provided with more training data, Fast R-CNN can improve its performance; Softmax works slightly better vs. post-hoc SVM; More proposals are harmful.

4.4 Disadvantages / Speed Bottleneck

Although Fast R-CNN is faster than R-CNN in both training and testing time, it still has certain issues. That is because it uses selective search to get regions of interest same as R-CNN, this is a slow and time-consuming process.

5. Faster R-CNN

Faster R-CNN was proposed by Ren et al. in 2016 [12]. In Fast R-CNN, the bottleneck of the architecture is selective search, which was replaced in Faster R-CNN by Region Proposal Network (RPN). Convolutional features of the image is shared with the detection network, which makes it nearly cost-free region proposals.

5.1 Faster R-CNN Model Workflow

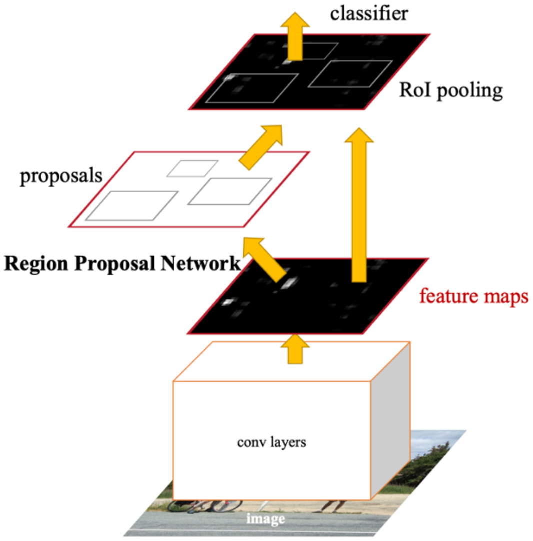


Figure 5: Faster R-CNN Architecture

Faster R-CNN consists two models – FCN and Fast R-CNN detector. The fully convolutional network will propose regions, Fast R-CNN detector will classify the proposed regions. The entire system is a single, unified network for object detection.

Step 1: Take an image as input and pass it to the CNN, then generates a feature map.

Step 2: Apply Region proposal network on the feature map, then returns a set of rectangular the object proposals and their 2k scores / objectness score.

Step 3: Apply RoI pooling layer on these proposals so that all proposals are of the same size.

Step 4: Pass these proposals to a fully connected layer, the softmax layer and linear regression layer will classify and output the bounding boxes for objects.

5.2 How Region Proposal Network (RPN) works

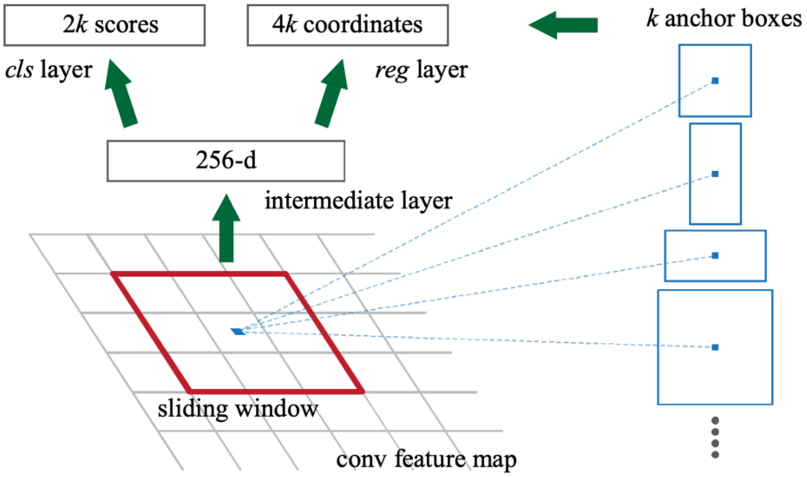


Figure 6: Region Proposal Network (RPN)

Region Proposal Network has a classifier and a regressor.

After getting the feature maps from CNN, RPN uses a sliding window over these feature maps, and at each sliding-window location, it generates *k* Anchor. See Figure 6. And by default three scales (128, 256, 512) and three aspect ratios(1:1, 1:2, 2:1) are picked for each anchor, so there will be *k* = 9 anchors at each sliding position. For a convolutional feature map of a size W x H, there are WH*k* anchors in total.

For each anchor, RPN predicts two things: First, the classifier doesn't consider which category the object belongs to, it only predicts the probability that an anchor is an object or not. An anchor is positive (object) if: (I) it has the highest Intersection-over-Union (IoU) overlap with ground-truth box, or (II) it has an IoU overlap higher than 0.7 with any ground-truth box. Note, if its IoU ratios for all ground-truth boxes are lower than 0.3, the anchor will be assigned as negative. Second, regression regresses the coordinates of the proposal to better fit the object.

5.3 Achievements / Advantages of Faster R-CNN

The key achievement of Faster R-CNN is RPN. It is an efficient and accurate region proposal generation method which makes Region proposal step nearly cost-free. This method enables a unified, deep-learning-based object detection system to run at 5-17 fps. In addition, region proposal quality and object detection accuracy are also improved.

5.4 Disadvantages of Faster R-CNN

Faster R-CNN is an end-to-end training system, the performance of the systems further ahead heavily depends on how the previous system performed. (This is only my personal opinion, not concluded from the paper.)

6. Mask R-CNN

Mask R-CNN, extends Faster R-CNN to pixel-level segmentation, was proposed by He et al. in 2017 [4]. Mask R-CNN efficiently detects objects within an image and generates a high-quality segmentation mask for each instance simultaneously.

6. 2 Mask R-CNN Architecture

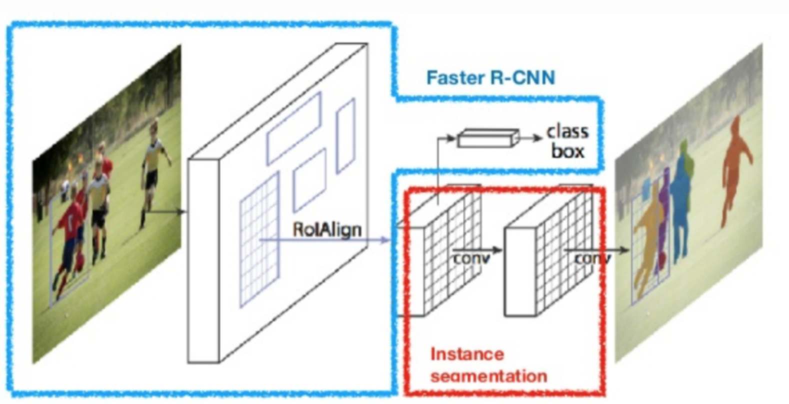


Figure 7: Mask R-CNN Architecture

Mask R-CNN is conceptually simple: Faster R-CNN has two outputs for each candidate object, a class label and a bounding-box offset; based on that, Mask R-CNN adds a third parallel branch to output the object mask.

Due to pixel-level segmentation requires more fine-grained alignment than bounding boxes, Mask R-CNN improves the RoI pooling layer to RoIAlign layer, so that the extracted features can be more precisely mapped to the regions of the original image at pixel-level.

6. 3 RoIAlign & Loss Function

RoIAlign layer is proposed to reduce the misalignment caused by quantization in RoI pooling. RoIAlign successfully reduces the harsh quantization so that the extracted features can be properly mapped to the original regions in pixel-level. Bilinear interpolation is applied to compute the exact values (no rounding) of the extracted features.

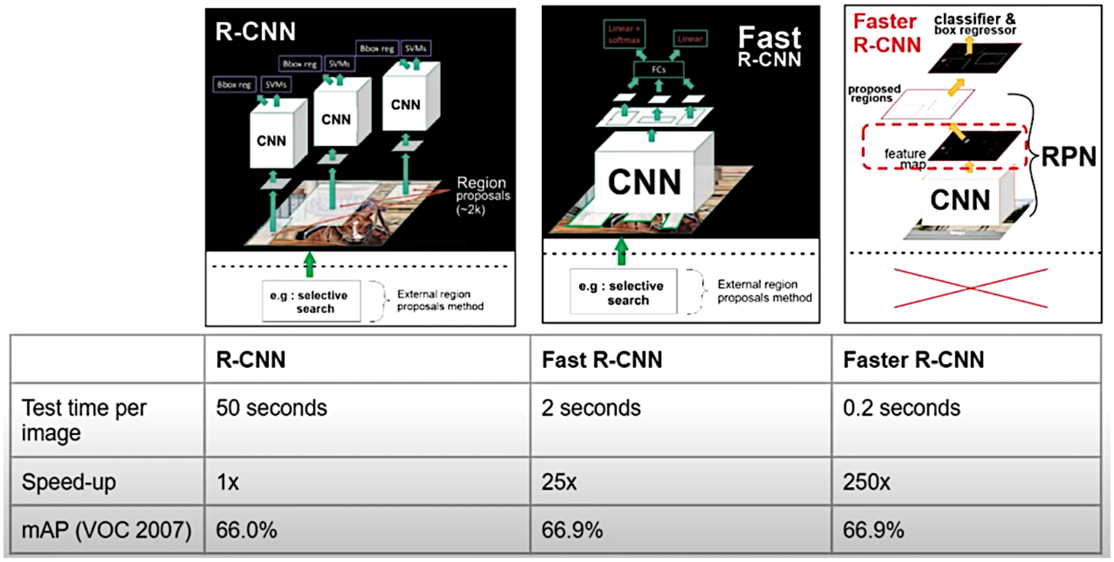
The loss of Mask R-CNN is a combination of classification, localization and segmentation mask: $L = L_{cls} + L_{box} + L_{mask}$, where L_{cls} and L_{box} are the same as in Faster R-CNN. The mask branch has a K m^2-dimensional output for each RoI, which encodes K binary masks of resolution m x m, one for each of the K classes. Thus, the total output is of size $K \cdot \mathrm{m}^2$.

6.4 Achievements / Advantages of Mask R-CNN

Mask R-CNN is simple to train and it’s a combination of FCN and Faster R-CNN detector. Mask R-CNN can run at 5 fps. And it is easy to recast Mask R-CNN to other tasks, for example, it allows us to estimate human poses in the same framework. More importantly, Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners [4].

Key improvements: (I) RoIAlign, (II) decoupling mask with class prediction that generates binary mask, and (III) parallel object detection and mask generator.

7. Compare R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN



	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image	50 seconds	2 seconds	0.2 seconds
Speed-up	1x	25x	250x
mAP (VOC 2007)	66.0%	66.9%	66.9%

Figure 8: Comparison between R-CNN, Fast R-CNN and Faster R-CNN

R-CNN generates the region proposals using selective search algorithm first and then computes features for each proposal using a large CNN. It takes R-CNN 50s to test one image. And because R-CNN involves three models separately without much shared computation , it takes 84 hours to train the model.

Fast R-CNN also applies an external selective search algorithm to find regions in the image, but it swaps the sequence of generating region proposals and the use of CNN, so the computation of convolutional layers among proposals for an image is shared, which successfully shorter the test time to 2s. And Fast R-CNN trains the whole system end-to-end all at once, so it only takes 9.5 hours (using VGG-16 CNN on PASCAL VOC 2007) to train the model. The mAP is improved as well compared to R-CNN.

Faster R-CNN does not use external region proposal method anymore, instead it inserts a Region Proposal Network (RPN) after the last convolutional layer, so this really bring down the test time per image to 0.2s, which is nearly cost-free.

Mask R-CNN uses the same basic architecture as Faster R-CNN, and addition to that, it adds a fully convolution layer to locate objects at the pixel level and further increase the accuracy of object detection.

Pascal VOC2007, VOC2007, and MSCOCO are three most commonly used datasets for evaluating detection algorithms. Pascal VOC2012 and VOC2007 are mid-scale datasets with 2 or 3 objects per image and the range of object size in VOC dataset is not large. For MSCOCO, there are nearly 10 objects per image and the majority object are small objects with large scale ranges [5]. R-CNN, Fast R-CNN, Faster R-CNN used these datasets to train and test, so the results should be convincing.

8. Conclusions

Object detection is one of the fundamental visual recognition problems in computer vision field and has been a hot research topic in recent years, and new state-of-the-art results have been reported almost every few months [10]. We can see the evolution from R-CNN to Mask R-CNN only took three years. After reading and studying through the related papers, I understand that advancements such as Mask R-CNN are the sum of intuitive, hard work and collaboration, its successful work is built upon the previous work.

No matter how advanced the technique develops, there are still some open challenges left in the field of object detection. For example, the mAP though is not low now, it still has the space to be improved; Visual object detection result can be affected by contexts, however, the effort and focus on utilize contextual information are very limited. It can be a promising future direction to incorporate contexts for object detection.

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[2] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[3] Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062.

[4] He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 2961-2969).

[5] Wu, X., Sahoo, D., & Hoi, S. C. (2020). Recent advances in deep learning for object detection. Neurocomputing.

[6] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. International journal of computer vision, 60(2), 91-110.

[7] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). IEEE.

[8] Lienhart, R., & Maydt, J. (2002, September). An extended set of haar-like features for rapid object detection. In Proceedings. international conference on image processing (Vol. 1, pp. I-I). IEEE.

[9] Bay, H., Tuytelaars, T., & Van Gool, L. (2006, May). Surf: Speeded up robust features. In European conference on computer vision (pp. 404-417). Springer, Berlin, Heidelberg.

[10] Zhao, Z. Q., Zheng, P., Xu, S. T., & Wu, X. (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.

[11] Girshick, R. (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

[12] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[13] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125).

[14] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[15] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger.

In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[16] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[17] Uijlings, J. R., Van De Sande, K. E., Gevers, T., & Smeulders, A. W. (2013). Selective search for object recognition. International journal of computer vision, 104(2), 154-171.

[18] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

[19] He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9), 1904-1916.

[20] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

[21] Tan, M., Pang, R., & Le, Q. V. (2020). Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10781-10790).

[22] Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

[23] Qin, X., Zhang, Z., Huang, C., Gao, C., Dehghan, M., & Jagersand, M. (2019). Basnet: Boundary-aware salient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7479-7489).

[24] Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.