Team 9: Object Detection, Recognition and Segmentation

Mar Ferrer Ferrer[1], Alex Tempelaar[2], Víctor Ubieto[3]

Universitat Politècnica de Catalunya, Spain
mar.ferrerf@e-campus.uab.cat[1], alexander.tempelaar@e-campus.uab.cat[2], victor.ubieto@e-campus.uab.cat[3]

Abstract. In this document it is presented the evolution of the most revolutionary techniques that appeared in Object Detection until now. Therefore, the most famous models, its innovations and problems are presented and discussed.

1 Introduction

Object detection, one of the most fundamental and challenging problems in Computer Vision, seeks to locate object instances from a large number of predefined categories in natural images. The spatial location can be defined using a bounding box.

The tasks related to the generic object detection can be divided into object classification, which aims to assess the presence of objects from a given set of object classes in an image without need of location; object recognition, which denotes the part of identifying/localizing all the objects in the image; and image segmentation, which is very similar to object detection as it to distinguish different instances of the same object class.

2 Evolution of the Model

The first approach of a model that started using deep networks for object detection was OverFeat in 2013 (see in section 3.1). It basically consists in the usage of the AlexNet model (state of the art at that time) but having two outputs, a softmax that gives the class and confidence of the prediction, and a regressor that gives the bounding box that localises the object in the image. It also recasted the AlexNet model by using a 1 by 1 convolution for the output, so the classifier could be applied to different input size images.

In the ends of the same year, R-CNN (section 3.2) appeared with a new approach, the usage of proposal regions. It eliminates most of the false positive for each proposal. The drawbacks were the training and inference time. But by that time the images had to be warped to fit the input size used in training because of the fully connected layer that was used at the end of the network. Nevertheless, in 2014, Spatial Pyramid Pooling model (section 3.3) provided a pooling layer that produced fixed length representations from variable size feature maps (inputs).

This advances were introduced in R-CNN to create Fast R-CNN in 2015 (section 3.4). Although it changed the Spatial Pyramid Pooling Layer for a Region of Interest (ROI) Pooling Layer which did practically the same. Moreover, Fast R-CNN allows an end-to-end training significantly reducing its computation time.

The same year, it was noticed that the region proposal model worked well in slow models. So they decided to also use another network only to predict region proposals instead of using traditional computer vision techniques (Selective Search algorithms). This new implementation was called Faster R-CNN (section 3.5).

Until this point, the evolution of the models increased both the mean Average Precision (mAP) and the speed. Nevertheless, the computation time was not low enough to be applied for realistic applications. The reason is that the detectors presented are all divided in two steps (two stash object detectors): the first stash is composed by the extraction of the convolutional features and object proposals, and the second stash is composed by the region level computations, where features and results are extracted for each proposal.

Therefore, in 2015, a novel model with real-time speed appeared. With the name of You Only Look Once (YOLO, section 3.6), it proposes an object detector architecture that works on a single stash (Single Shot Object Detector). The key idea is that we extract a feature map, and each cell of the map makes a prediction of a bounding box. The drawback of having real-time predictions is the worse mAP results when comparing with Fast R-CNN or Faster R-CNN, this is the trade-off problem between time and and mAP so depending on the application it is chosen to use one or two stashes.

In late 2016, another model called Single Shot Multibox Detector (section 3.7) was created very similar to YOLO. In this model, predictions are taken at different feature maps, consequently, a big amount of detections are obtained. Therefore, it is required to filter the ones with low confidence, this can be achieved using Non-Maximum Suppression.

From now on, alternative ideas have lead to the creation of more new models like Feature Pyramid Networks (FPN), RetinaNet, and Mask R-CNN (all in 2017). Each of them providing a novel idea to improve the overall performance of the model and its applications (these models are detailed in sections 3.8, 3.9, and 3.10 respectively).

Lately, there have appeared some new approaches such as EfficientDet for the case of a single stash [1], or Cascade Mask R-RCNN for the two-case stash [2], outperforming the models mentioned until now [3][4].

3 Related Work

In this section are presented a brief description of all the networks mentioned in the evolution of the model of Object Detection and some of the current state of the art.

3.1 OverFeat

OverFeat uses a single shared Convolutional Network for learn and solve simultaneoslu classification, localization and detection problems. The work presents the implementation of a multiscale sliding window within a ConvNet, and a novel deep learning approach to localization by learning to predict object boundaries [5].

3.2 R-CNN

This model combines regions proposals with CNNs. The approach combines two things: the usage of CNNs to bottom-up region proposals to localize and segment objects, and the performance boost obtained by using supervised pre-training [6].

3.3 Spatial Pyramid Pooling: SPP

The main contribution of this paper is the Spatial Pyramid Pooling Layer which allows to calculate fixed length representations for all the region proposals, independently on their sizes. It trains faster than R-CNN but is still slow. However, in inference time, the improvement is substantial (100 times faster than R-CNN) [7].

3.4 Fast R-CNN

One of the main changes in this model is that Fast R-CNNs allow an end-to-end training, since the Support Vector Machine used for computing the class confidence is substituted for a Softmax [8]. Also, it has a Region Proposal step that uses Selective Search, which greedily merges superpixels based on engineered low-level features. Moreover, it includes a Region of Interest (ROI) pooling which follows the same idea than the Spatial Pyramid Layer but instead of using a pyramid, and adaptive pooling is able to adapt to any size of the Region Proposals output. The training is faster than with SPPs and the inference is more or less the same.

3.5 Faster R-CNN

Until the creation of this model, object detection networks used Selective Search algorithm to hypothesise object location. Fast R-CNN improved the inference step and was faster than the previous models, nevertheless, the Selective Search is the bottleneck of the model. Faster R-CNN proposes to train a deep network to be in charge of the Region Proposal task: Region Proposal Network (RPN). Other versions of this model were created where they experimented with different backbones, giving ResNet the best results [9].

RPN The input is the feature map of the last convolutional layer and for every cell of this feature map it makes k predictions for k different anchor boxes (anchor boxes are boxes that have specific sizes and shapes in order to be representative of the data in the training set).

3.6 You Only Look Once: YOLO

YOLO proposes a single neural network that predicts bounding boxes and class probabilities directly from full images in one step. Because of the one stash pipeline, it can be optimized end-to-end directly on detection performance which makes it really fast [10]. Over the years, new versions of YOLO have been created improving the overall score.

3.7 SSD: Single Shot Multibox Dectector

This approach, discretizes the output space of bounding boxes into default boxes with different aspect rations and scales. When predicting, the network generates scores for each object category in each default box [11].

3.8 Feature Pyramid Network: FPN

Feature Pyramid Network architecture shows significant improvement as a generic feature extractor in several applications. In this model, they exploit the multi-scale, pyramidal hierarchy of deep CNN to construct feature pyramids with marginal extra cost [12].

3.9 RetinaNet

This model proposes a single stash object detector that performs almost equally well in mAP than two stash object detectors. In this research, they focus on solving the reason of the under-performance of one stash methods, and correct it by reshaping the standard cross entropy loss such that it down-weights the loss assigned to well-classified examples [13].

3.10 Mask R-CNN

This model is an evolution of the Faster R-CNN, and basically solves two tasks at the same task, object detection and object segmentation (third output). The idea is that solving one task helps the other one, for instance if the model improve at detecting, it will consequently get better at segmenting objects, and otherwise [14].

References

1. M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," 2020.
2. Y. Liu, Y. Wang, S. Wang, T. Liang, Q. Zhao, Z. Tang, and H. Ling, "Cbnet: A novel composite backbone network architecture for object detection," 2019.
3. T. Hoeser and C. Kuenzer, "Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends," *Remote Sensing*, vol. 12, no. 10, 2020. [Online]. Available: https://www.mdpi.com/2072-4292/12/10/1667
4. L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," 2019.
5. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2013. [Online]. Available: http://arxiv.org/abs/1312.6229
6. R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation." IEEE Computer Society, 9 2014, pp. 580–587. [Online]. Available: http://www.cs.berkeley.edu/~rbg/rcnn.
7. K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8691 LNCS, pp. 346–361, 6 2014. [Online]. Available: http://arxiv.org/abs/1406.4729 http://dx.doi.org/10.1007/978-3-319-10578-9_23
8. R. Girshick, "Fast r-cnn," 2015.
9. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 6 2017. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/results
10. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.
11. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9905 LNCS, pp. 21–37, 12 2015. [Online]. Available: http://arxiv.org/abs/1512.02325 http://dx.doi.org/10.1007/978-3-319-46448-0_2
12. T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," 12 2016. [Online]. Available: http://arxiv.org/abs/1612.03144
13. T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 318–327, 8 2017. [Online]. Available: http://arxiv.org/abs/1708.02002
14. K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, pp. 386–397, 2 2020. [Online]. Available: https://github.com/