Object Detection Overview

Carlo M. De Masi
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
carlo.maria.demasi@ijs.si

Mitja Luštrek
Jožef Stefan Institute
Jamova cesta 39, 1000 Ljubljana, Slovenia
mitja.lustrek@ijs.si

ABSTRACT
In this paper, we present a brief review of some of the main algorithms adopted in the field of computer vision for the aim of object detection. We highlight the working principles of two main families of models, Region-based Convolutional Neural Network detectors ("R-CNN") and Single-Shot object detectors (SSD, YOLO), and present a comparison between them, with a focus on the trade-off between speed and accuracy and its dependence on various models parameters.

Keywords
computer vision, object detection, CNNs

1. INTRODUCTION
Most computer vision tasks can be roughly divided into three main categories [10]:

- **image classification**: determining whether a certain class of objects is present in the image or not (Fig. 1a);
- **object detection:** determining that an object belonging to a certain class is present, and localizing it within the image (Fig. 1b);
- **semantic scene labeling**: classification of each pixel of the image as belonging to a certain class. Individual instances of the same object are usually not segmented (Fig. 1c), but modern datasets [10, 9] provide labels to distinguish between them (Fig. 1d).

In less than a decade, techniques to solve these tasks have undergone a period of extremely rapid development, most notably fueled by the successful employment of Convolutional Neural Networks (CNN) in the field.
The adoption of CNN-based algorithms was kick-started by the paper by Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton [8]. This paper showed the effectiveness of CNNs in computer vision problems, and introduced a number of techniques that are still used today, such as the adoption of ReLU as activation functions and the use of data augmentation techniques. The standard structure of CNN-based models, basically consisting of simple stacking of convolutional and pooling layers followed by one or more fully-connected layers, has been successfully used for some time in order to solve localization, object detection and human pose estimation problems [17, 20, 1, 4, 21, 19]. Improvements were generally achieved by increasing the depth and width of the models, and by using larger amount of training data, at the cost of increasing the needed computational power and the risk of over-fitting [19]. This basic paradigm has been challenged by Google with the introduction of the "Inception" architecture [19], where parts of the network are not located sequentially, so that different operations such as pooling or convolution can be performed in parallel.
Among the many proposed architectures (see also [5, 24]), arguably one of the most impactful contributions to object detection has been given by the introduction of the Region-based Convolutional Neural Network detectors ("R-CNN") [4] and its modification (Fast R-CNN and Faster R-CNN [3, 15]). The accuracy of these detectors, however, comes at a computational cost, which can be reduced by the adoption of the so-called Single Shot detectors.
In this paper, we present an overview of the region-based and single-shot families of models. In Section 2, we describe a few methods to identify Regions of Interest (RoIs), i.e. areas of the picture where objects are located, which is a fundamental ingredient for region-based detection algorithms. Sections 3 and 4 are dedicated to region-based and single-shot detectors, respectively. In section 5, we summarize a comparison of the performances of these algorithms, and how their trade-off between speed and accuracy is influenced by different factors.

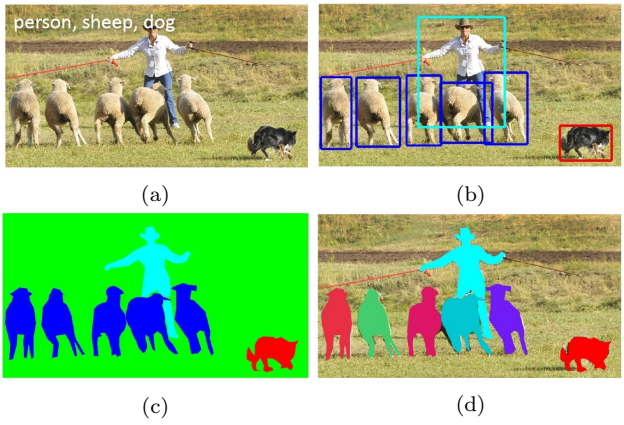


Figure 1: From [10], different types of computer vision tasks. Image classification (a), object detection (b), and semantic scene labeling (c,d).

2. REGIONS OF INTEREST (ROI) SELECTION ALGORITHMS

Region-based methods generally tackle object detection tasks by dividing it into two steps. First, it is necessary to locate the so-called Regions of Interest (RoI), i.e. areas of the image where possible objects are located. After this, the classification step takes place, where the previously located areas are classified. RoIs whose content is classified as belonging to one of the considered classes within a certain confidence are detected.
In the following, we make a short description of two of the main methods adopted to identify RoIs.

2.1 Selective search

Region proposal algorithms identify RoIs by adopting image segmentation techniques, which consist of grouping pixels based on their similarity according to some criteria. A commonly adopted method is Selective Search (SS), where groups of similar regions are created hierarchically, starting from single pixels, based on color, texture, size and shape compatibility [22]. A general important requirement for region proposal methods is that they should have a very high recall; false positives can then be rejected in the following classification phase.
Fig. 2, from [22], shows an example of the process.

Figure 2: From [22]. All possible ROIs during the merging.

2.2 Region Proposal Network (RPN)

Region proposal networks (RPN) are the distinctive feature of Faster R-CNNs detectors (see Section 3), and they eliminate the need to use an external algorithm for selecting RoIs. In this method, the feature map produced by the first convolutional layer of the detector is passed to a CNN that produces region proposals by predicting their bounding boxes and "objectness" scores, measuring whether each box contains an object or not [15].

3. REGION-BASED OBJECT DETECTORS (R-CNN, FAST R-CNN, FASTER R-CNN)

As mentioned earlier, region-based object detectors work in two steps, consisting of the determination of RoIs and their following classification.
The first suggested implementation of this algorithm, **R-CNN** [4], employs a region proposal method to create $\approx$ 2000 ROIs. As shown in Fig. 3 (top left panel), each of the identified RoIs is then resized and fed as input to a CNN, followed by fully connected layers to classify the object and refine the boundary box.
Even though the R-CNN algorithm is very accurate, it has also the downside of being quite slow. The high number of proposals makes the algorithm slow, since each RoI is processed by the CNN separately, which means that the whole feature extraction process is repeated 2000 times.
This limitation has been overcome in **Fast R-CNNs** [3], as depicted in Fig 3 (bottom left panel). In this architecture, features are extracted only once for the whole image by using a CNN, while an external region proposal method (such as selective search) is used to create RoIs. After this step, the feature map and the RoIs are combined, producing patches that, as in the previous case, are resized (RoI pooling layer) and passed as input to a fully connected layer for the object detection.
In the case of Fast R-CNNs, the main bottleneck is caused by the use of the external region proposal method, which usually runs on a CPU and is slower than the rest of the process; out of the 2.3 seconds needed by Fast R-CNN to make a prediction in testing, $\approx$2 seconds are used for generating the 2000 ROIs [3]. So, a further improvement in the algorithm speed has been provided with the introduction of **Faster R-CNN** [15], by substituting the external region proposal method with the convolutional Region Proposal Network (RPN) we presented in Sec. 2.2 (see Fig. 3, right panel).

4. SINGLE-SHOT OBJECT DETECTORS

Differently from region proposal detectors, which perform region proposals and region classifications in two steps, single-shot detectors simultaneously predict bounding boxes and the class as they processes the image in one shot.

4.1 SSD

SSD [11] is one of the fastest object detectors available; its two variants, SSD300 and SSD512, achieved up to 74.3% mAP at 59 FPS and 76.9% mAP at 22 FPS, respectively, on the VOC2007 [2] test dataset.
The working principles of SSD can be summarized as follows:

- the image is passed through a series of convolutional layers, thus producing several sets of feature maps at different scales (4x4, 8x8, etc.);
- a pre-defined, default set of bounding boxes (similar to the "anchors" in RPN [15]) of different aspect ratios is provided for each location in all the produced feature maps;
- for each of these default boxes, both the offsets to the ground truth boxes and the confidence for all classes are predicted;
- default boxes are matched to ground truth boxes based on IoU (Intersection Over Union, [23]). The best predicted box is labeled a positive, along with all other boxes that have an IoU with the truth > 0.5 (see Fig. 4).

The downside of skipping region proposal is that SSD draws and classifies bounding boxes of many shapes and scales in every single position in the image, so that most of them are negative examples. For this reason, highly-overlapping

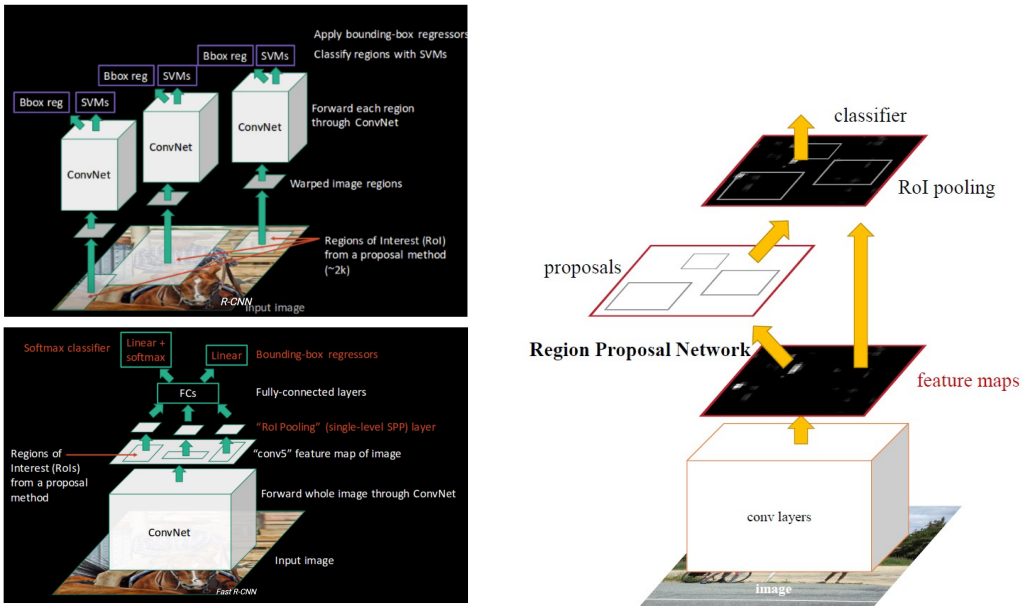
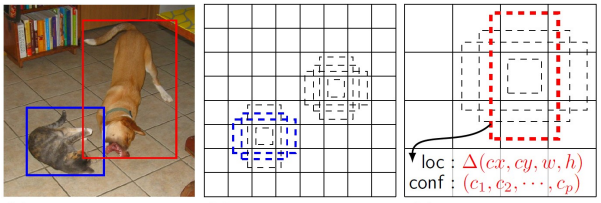


Figure 3: From `http://www.robots.ox.ac.uk/~tvg/publications/talks/fast-rcnn-slides.pdf`: model scheme for R-CNN (top left), Fast R-CNN (bottom left), Faster R-CNN (right).



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

Figure 4: From [11], SSD framework. Out of the many default boxes, two are matched with the cat and one with the dog, which are treated as positives and the rest as negatives

boxes are grouped into a single one ("non-maximum suppression", [16]). Moreover, the high number of negatives leads to a significant imbalance between negative and positive samples for training, which is overcome by only using the ones with highest confidence loss (a part of the overall loss function, measuring how confident the network is of the "objectness" of the box), so that the ratio between the negatives and positives is at most 3:1 ("hard negative mining").

4.2 YOLO (You Only Look Once)

As in SSD, YOLO [12, 13, 14] uses a single neural network for detection (Fig. 5).

The input image is divided into a grid of SxS cells, and B bounding boxes are produced for each cell. For each box, a score is calculated, indicating the confidence for that box to contain an object (of any class) and the accuracy of the box in terms of its IoU with the ground truth ($Pr(Object) \times IOU_{pred}^{truth}$). For each cell, C (C = number of possible classes) probabilities calculated, conditioned on the grid cell itself containing an object. At test time, these conditional class probabilities and individual box confidence predictions are multiplied, in order to obtain class-specific confidence scores for each box.

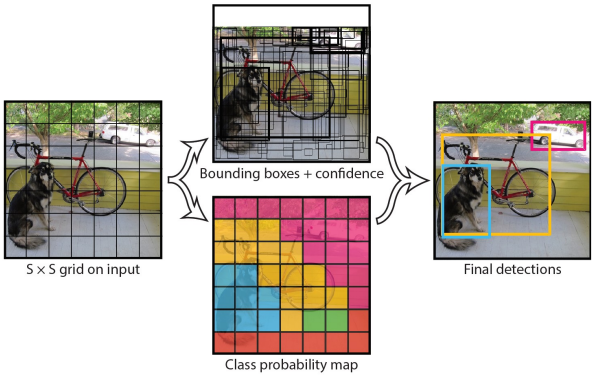


Figure 5: From [11], SSD framework. Out of the many default boxes, two are matched with the cat and one with the dog, which are treated as positives and the rest as negatives

5. COMPARISON

A performance comparison between the various presented algorithms, although being of great interest, can be tricky. Standard metrics like mAP [2] do not take into account factors like time and memory usage [7], which are of vital importance when real-time performance is required. On the other hand, greater speeds are obtained by sacrificing some accuracy, and it is important to be aware of the mechanisms influencing this trade-off. Finally, results reported by the various papers are generally obtained by using different settings, which makes their comparison less (if at all) significant. A work by Google research [7] offers a survey to study the trade-off between speed and accuracy for a series of models, including Faster R-CNN and SSD. The various presented models have been re-implemented in TensorFlow and trained on the MS COCO dataset. The effect of adopting different feature extraction architectures (MobileNet [6], VGG-16 [18], Inception, etc.) for each of the models has

also been tested. In the following, we briefly summarize their main results.

- SSD models are faster on average, but cannot beat the Faster R-CNN in accuracy. Faster R-CNN requires at least 100 ms per image (Fig. 6), while SSD with MobileNet as feature extractor provides the best accuracy tradeoff within the fastest detectors. The highest accuracy is achieved by Faster R-CNN using Inception ResNet as feature extractor with 300 proposals, running at 1 second per image;
- Choice of feature extractors impacts detection accuracy for Faster R-CNN, but it is less important for SSD;
- For large objects, SSD can outperform Faster R-CNN in accuracy with faster extractors, but its accuracy drops significantly for smaller objects;
- Input image resolution strongly impacts performance: on average, reducing image size by half lowers accuracy by 15.88% and inference time by 27.4%;

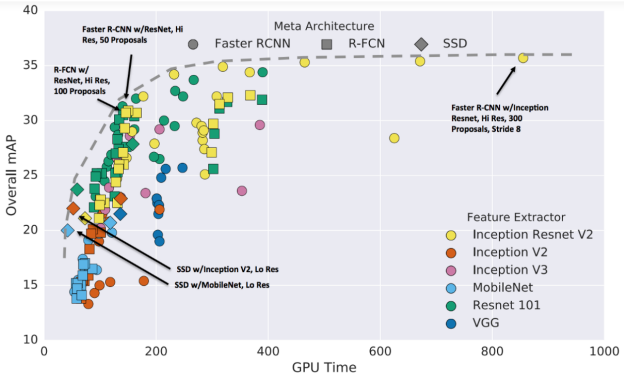


Figure 6: From [7], accuracy vs time. Different marker shapes indicate meta-architecture, while colors indicate feature extractor.

6. CONCLUSIONS

In this paper, we presented an overview of the region-based (R-CNN, Fast and Faster R-CNN) and of the single shot (SSD, YOLO) families of algorithms for object detection. An analysis of the speed and accuracy of the model parameters, and of how they are influenced by the choice of the models parameters, is briefly summarized.

7. REFERENCES

[1] D. Erhan, C. Szegedy, and A. Toshev. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2147–2154, 2014.

[2] M. Everingham, L. Van Gool, C. K. Williams, et al. *International journal of computer vision*, 88(2):303–338, 2010.

[3] R. Girshick. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[4] R. Girshick, J. Donahue, T. Darrell, et al. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[5] K. He, X. Zhang, S. Ren, et al. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. *arXiv preprint arXiv:1704.04861*, 2017.

[7] J. Huang, V. Rathod, C. Sun, et al. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7310–7311, 2017.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[9] A. Kuznetsova, H. Rom, N. Alldrin, et al. *arXiv preprint arXiv:1811.00982*, 2018.

[10] T.-Y. Lin, M. Maire, S. Belongie, et al. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

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

[13] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[14] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[15] S. Ren, K. He, R. Girshick, et al. In *Advances in neural information processing systems*, pages 91–99, 2015.

[16] R. Rothe, M. Guillaumin, and L. Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian conference on computer vision*, pages 290–306. Springer, 2014.

[17] P. Sermanet, D. Eigen, X. Zhang, et al. *arXiv preprint arXiv:1312.6229*, 2013.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] C. Szegedy, W. Liu, Y. Jia, et al. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[20] C. Szegedy, A. Toshev, and D. Erhan. In *Advances in neural information processing systems*, pages 2553–2561, 2013.

[21] A. Toshev and C. Szegedy. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014.

[22] K. E. Van de Sande, J. R. Uijlings, et al. In *ICCV*, volume 1, page 7, 2011.

[23] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola. 2019. `http://www.d2l.ai`.

[24] K. Zhang, M. Sun, T. X. Han, et al. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(6):1303–1314, 2017.