An Overview of Deep Learning-Based Object Detection Methods

Yassine Bouafia

Mostafa Ben Boulaid University
Batna, Algeria
bouafia.yassine.sim@gmail.com

Larbi Guezouli

Mostafa Ben Boulaid University
Batna, Algeria
larbi.guezouli@univ-batna2.dz

***Abstract*—In recent years, there has been rapid development in the research area of deep learning. Deep learning was used to solve different problems, such as visual recognition, speech recognition and handwriting recognition and was achieved a very good performance. In deep learning, Convolutional Neural Networks (ConvNets or CNNs) are found to give the most accurate results in image recognition and object detection problems.**

In this paper we'll go into summarizing some of the most important deep learning models used for object detection tasks over this last recent year, since the creation of AlexNet in 2012. Then, we'll make a comparison in speed and accuracy between the most used state-of-the-art methods in object detection.

Keywords—Object Detection, Deep Learning Methods, Convolutional Neural Networks

I. Introduction

Object detection is one of the most active field of research in computer vision, where it involves both object classification, classifying every object in the image and object localization, localizing each object by drawing a bounding box around it. Today with the continuous increase in the use of object detection in several interesting applications such as video surveillance, robotic, self-drive car, etc. it became necessary to develop more accurate and faster systems. Deformable Part Model [1] was the dominant detection framework before the widespread use of Convolutional Neural Networks. Recently, Convolutional Neural Networks contributed to a significant increase in the accuracy of object detection and greatly surpassed other classic models such as Viola & Jones framework [2], and Histograms of Oriented Gradient (HoG) [3].

The rest of the paper is organized as follows. Firstly, Section II presents challenges and problems to build an ideal detector. Then, Section III provides a brief history of Convolutional Neural Networks. Next, Sections III presents set of datasets for object recognition. After that, Section IV offer an overview of a set of most important object detection methods and classifiers during the past few years. Then in section V, we make a comparison between set of methods in accuracy and speed. Finally, section VI concludes the overview.

II. Challenges and Problems

An ideal detector should have:

A. High accuracy in localization and recognition:

The detector must be able to locate and recognize objects in images accurately.

B. High efficiency in time and memory:

The detection task should run at a sufficiently frame rate with acceptable memory and storage usage.

For accuracy, we have two main challenges:

- Firstly, intra-class variations, where each object category can have many object instances. These instances varying in several features like color, texture, size, shape and different poses in case of non-rigid classes. The variations are caused by changes in a set of factors such as locations, weather conditions, cameras, backgrounds, illuminations, viewpoints, and distance. Further challenges can be added such as illumination, pose, scale, occlusion, background clutter, shading, blur, motion, noise corruption and poor resolution.
- In addition to intra-class variations, we have huge number of object categories in real world, where the number of object categories in existing benchmark datasets is much smaller than that can be recognized by humans.

For efficiency, the challenge is the need to detect objects in real time. This often requires big performance or sacrificing accuracy versus speed. On the other hand, we need to build an efficient detector that work in devices that have limited computational capabilities and storage space such as mobiles.

III. History of Convolutional Neural Networks

Convolutional Neural Networks is a deep learning architecture that have proven very effective in computer vision tasks. CNN was inspired from the cat's visual cortex. In 1962, Hubel and Wiesel's [4], found that cells in animal visual cortex are responsible for detecting light in receptive fields.

Inspired by this discovery, Kunihiko Fukushima proposed a hierarchical model called Neocognitron[5]. Then, the first CNN was proposed by Hecht-Nielsenand LeCun et al., after many previous successful iterations since the year 1988, they developed a multi-layer artificial neural network trained with the backpropagation algorithm [6] called LeNet-5 [7] and it was used to classify handwritten digits. After this period the search in Deep Learning has entered a dark time. The next step for deep learning took place in 1999 owing to GPUs that make computers faster. Another big step was in 2009 when professor Fei-Fei Li launched ImageNet, a free data base of more than 14 million labeled images. With a large amount of data and the advent of GPUs, the field of CNN has gone through a renaissance phase and many publications have developed more efficient methods of training neural networks using GPU computing. In 2012 Krizhevsky, Ilya Sutskever, and Geoffrey Hinton won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) with deep CNN model called AlexNet, which was the beginning of a modern history of object detection.

IV. Datasets for object detection

Datasets play a very important role in object detection research, they have been one of the most important factors for the progress in the field, unfortunately data is harder and more expensive to generate. Over the last decade, a number of datasets have been made public to evaluate object detection algorithms. These datasets are collected from different scenarios and can therefore be used as a reference for applications. Below in TABLE I, there are a set of the popular datasets for object recognition.

TABLE I. OBJECT DETECTION DATASETS

Dataset	Total Images	Categories	Image Size	Started Year
MNIST[8]	60,000	10	28x28	1998
ImageNet[9]	>14 Millions	21841	500x400	2009
Caltech-101[10]	9,145	101	300x200	2004
Caltech-256 [11]	30,607	256	300x200	2007
MS COCO[12]	>328,000	91	640x480	2014
PASCAL VOC(2012)[13]	11,540	20	470x380	2005
CIFAR-10[14]	60,000	10	32x32	2009
Scene-15[15]	4,485	15	256x256	2006
Tiny images [16]	>79 Millions	53,464	32x32	2006
SUN[17]	131,072	908	500x300	2010
Open Images [18]	>9 Millions	>6000	varied	2017

V. Object detection methods based on deep learning

Currently we can organize object detectors in two main categories Fig. 2:

A. Two-stage detectors

Such as Faster R-CNN that divides the detection process in two steps. The first step uses a Region Proposal Network to generate regions of interests that have a high probability of being an object. The second step then performs the final classification and bounding-box regression of objects by taking these regions as input. These two steps are named the Region Proposal Step and the Object Detection Step respectively. Such models reach the highest accuracy rates, but are typically slow.

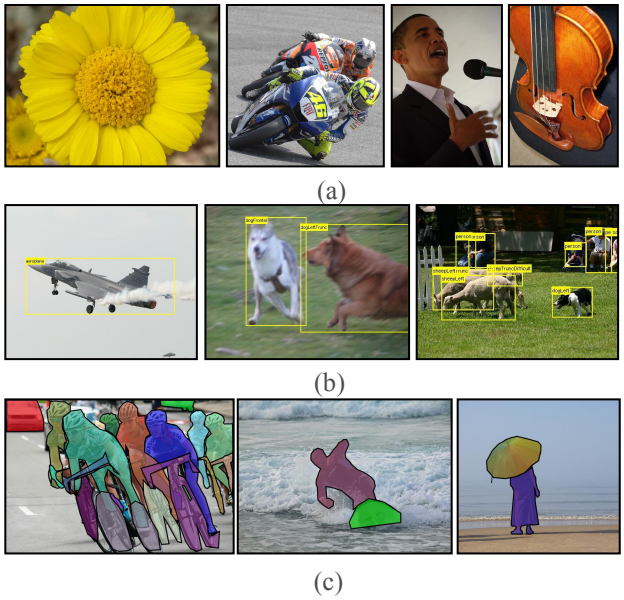

Fig. 1. Some images example from ImageNet Dataset (a), Pascal VOC Dataset (b) and COCO Dataset (c)

B. One-stage detectors

Such as YOLO and SSD, that treat object detection as a simple regression problem by taking an input image and learning the class probabilities and bounding box coordinates. The approach is simple and elegant because it completely eliminates region proposal generation, encapsulating all computation in a single network. Such models reach lower accuracy rates, but are much faster than two-stage object detectors and shown higher memory efficiency.

In this section we will show some of the most prominent detectors in recent years, and five of famous neural networks classifiers that have served as backbone in a lot of object detectors architectures. Most of these classifiers are trained in ImageNet dataset. All methods are listed in Fig. 3:

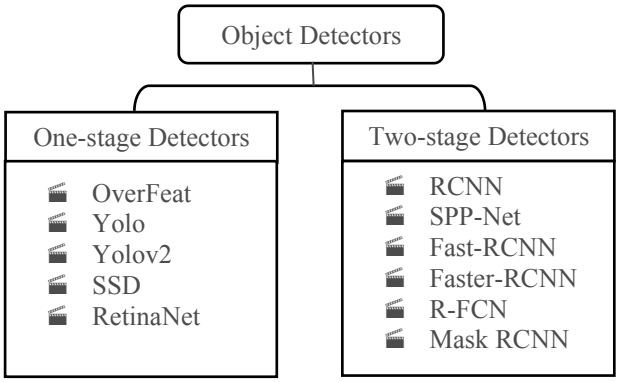

Fig. 2. Main categories of object detectors

A. Neural networks classifiers

*1) **AlexNet [19]:** is CNN for image classification created by A. Krizhevsky, I. Sutskever, and G. Hinton that was won the ILSVR 2012[20]competition and achieved top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry. Alexnet is neural network with five convolutional layers some of which are followed by max-pooling layers, it uses three fully-connected layers in the end of the network. The output of these layers feeds a final 1000-way softmax. Also AlexNet integrated various regularization techniques, such as data augmentation, dropout and used ReLU for the nonlinearity functions to decrease training time.*

*2) **ZFNet [21]:** Was the winner of the ILSVRC 2013 competition with 11.2% error rate. M. Zeiler and R. Fergus from NYU built neural network similar to AlexNet architecture with some modifications (7x7 kernel instead of 11x11 to retain more information) and a new visualization technique named Deconvolutional Network (deconvnet). This technique does the opposite work of convolution layer (from feature map to pixels). DeconvNet helps to examine different feature activations and their relationship to the input space.*

*3) **VGGNet [22]:** Simonyan and Zisserman of the University of Oxford created a 19 layer CNN that strictly used 3x3 filters with stride and padding of 1, along with 2x2 maxpooling layers with stride 2. VGGNet increased the depth of the network by adding more convolutional layers and taking advantage of very small convolutional filters in all layers. It was demonstrated that the representation depth was beneficial for the classification accuracy. Although rank 2 in ILSVRC 2014 which achieved 7.32% it is widely used as backbone in many object detectors architecture for extracting features from images.*

*4) **GoogleNet(Inception)[23]:** Is the winner of ILSVRC 2014 with 6.7% top 5 error rate. Their architecture consisted of 22 layers deep when counting only layers with parameters (or 27 layers if we also count pooling). Instead of traditionally stacking up conv and maxpooling layer sequentially, it stacks up Inception modules, which consists of multiple parallel conv and maxpooling layers with different kernel sizes. It uses 1x1 conv layer to reduce the depth of feature volume output.*

*5) **ResNet [24]:** Residual Neural Network won the ILSVRC 2015 competition with an unbelievable 3.6% error rate (human performance is 5-10%). ResNet is a new 152 layers network architecture with skip connections and features heavy batch normalization. In this technique they were able to train a very deep neural network with 152 layers. Instead of transforming the input representation to output representation, ResNet sequentially stacks residual blocks, each computes the change it wants to make to its input, and add that to its input to produce its output representation. This is slightly related to boosting.*

B. Neural networks detectors:

*1) **Overfeat [25]:** Is a sliding window approach that can be uses for classification, localization and detection. Overfeat using Convolutional Networks that contains eight layers. Five convolutional layers in the first and the remaining three are fully-connected layers. The output of these layers feeds a softmax layer to make prediction probability of 1000 classes. In the ILSVRC 2013 dataset, OverFeat ranked 4th in classification with 14.2% error, 1st in localization with 29.9% error (top 5 error rate) and 1st in detection established a new state of the art with 24.3% mean Average Precision (mAP).*

*2) **R-CNN[26]:** Regions with CNN features or R-CNN built by Ross Girshick et al. achieves 53.7% mAP on PASCAL VOC 2010, and 31.4% mAP on the ILSVRC2013 detection dataset. This results are considered as a large improvement over OverFeat network. R-CNN takes an input image, extracts around 2000 bottom-up region proposals using Selective Search [27], algorithm then computes features for each region proposal using convolution neural network, then classifies each region using linear SVMs.*

*3) **SPPNet [28]:** Spatial Pyramid Pooling Net is essentially an enhanced version of R-CNN by introducing two important concepts: adaptively-sized pooling. It uses spatial pooling after the last convolutional layer as opposed to traditionally used max-pooling, and computing feature volume only once. SPPNet ranked 3ed among all 38 teams attending ILSVRC 2014 with 8.06% error rate.*

*4) **Fast R-CNN [29]:**Takes as input an image and a set of object proposals (generated using selective search). After that, R-CNN applied convolution neural network to the entire image to produce a feature map, then, for each region proposals, used Region of Interest (RoI) Pooling on the feature map to extract features vector. Each features vector feeds a sequence of fully connected layers with a final feed forward network with two output layers: one for classification (produces class probability) and another for regression (produces bounding-box values). Fast R-CNN achieved top accuracy on PASCAL VOC 2012 with a mAP of 66%.*

*5) **Faster R-CNN [30]:** Slowest part in Fast R-CNN was*

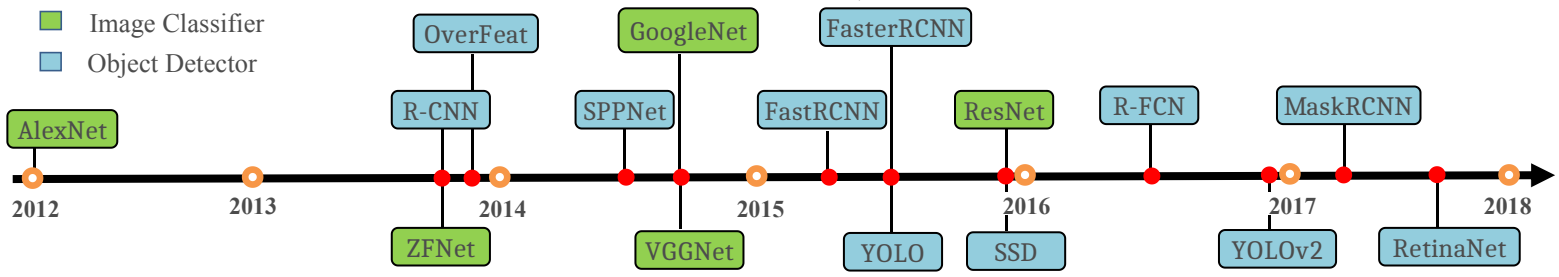

Fig. 3. Chronology of object detectors and classifiers based on the point in time of the first arXiv version

Selective Search or Edge boxes [31]. Faster R-CNN replaces selective search by a very small convolutional network called Region Proposal Network (RPN) after the last convolutional layer to generate regions of Interests. From that stage, the same pipeline as R-CNN is used region of interest (RoI) pooling, fully connected layer (FC), and then classification and regression heads. Faster R-CNN introduces the idea of anchor boxes to handle the variations in aspect ratio and scale of objects. Faster R-CNN achieves state-of-the-art object detection accuracy on PASCAL VOC 2007 (73.2% mAP) and 2012 (70.4% mAP) using 300 proposals per image.

*6) **YOLO[32]:** (You Only Look Once)YOLO transform detection as a regression problem it looks at the complete image at once as opposed to looking at only a generated region proposal in the previous methods. It uses a single convolutional neural network that contain 24 conv layers followed by 2 FC layers for both classification and localization tasks. YOLO divides the input image into S×S grid cells each cell have B anchors and it is responsible for predict B boxes and class probability for each box. The predicted bounding box consist of 5 values x, y, w, h and the confidence for those boxes, where (x, y) represent the center of the box relative to the bounds of the grid cell and w, h represent width and height relative to the whole image. This model allowing real time object detection (45 frames per second) and achieves a mAP of 63.4% on the VOC 2007 test set.*

*7) **Fast YOLO (Tiny YOLO)[32]:** Is a smaller version of YOLO with 9 convolutional layers instead of 24. It is much faster (runs at more than 155 fps) but less accurate than the normal YOLO model (57.1% mAP). Fast YOLO is the best solution when the detector speed is critical.*

*8) **SSD [33]:** Like YOLO, SSD (Single Shot Detector) is a method for detecting objects in images using a single deep neural network for both tasks of object localization and classification. It was released by C. Szegedy et al. at the end of November 2016 and reached new records in terms of performance and precision for object detection tasks, scoring over 74% mAP at 59 frames per second on standard datasets such as PascalVOC and COCO.*

*9) **R-FCN [34]:** Is a region-based, fully convolutional network for accurate and efficient object detection. In Faster RCNN after the RPN stage, each region proposal had to be cropped out and resized from the feature map and then fed into the Fast RCNN network. This step is the most time consuming .The R-FCN is an attempt to make the the network faster by making it fully convolutional and delaying this cropping step, the idea is increase speed by maximizing shared computation. As result R-FCN show competitive results on the PASCAL VOC 2007 datasets with 83.6% mAP. Meanwhile, is achieved at a test-time speed of 170ms per image, which is faster than Faster R-CNN.*

*10)**YOLOv2[35]:** After various improvements to the YOLO standard detection tasks like PASCAL VOC and COCO. YOLOv2 offered an easy trade-off between speed and accuracy. At 67 FPS, YOLOv2 gets 76.8 mAP on VOC 2007. At 40 FPS, YOLOv2 gets 78.6 mAP. YOLOv2 is real-time object detection system that can detect over 9000 object categories.*

*11)**Mask R-CNN [36]:** Running at 5 fps, it was built by the Facebook AI research team (FAIR) in April 2017 this approach added to RCNN a branch to predict an object mask. Mask RCNN consists of two stages. The first stage, proposes candidate object bounding boxes where there might be an object. Second, it predicts the class of the object, refines the bounding box and generates a mask in pixel level of the object based on the first stage proposal.*

*12)**RetinaNet [37]:** The Facebook AI research team design and train a simple detector called RetinaNet.It is a one-stage object detector which use new loss function called Focal loss instead of cross-entropy loss function. This new loss function has significantly increased the accuracy. The results show that when trained RetinaNet with the focal loss, we have one stage object detector that is able to match the speed of previous one-stage detectors and the accuracy of more complex two-stage detectors.*

VI. Comparison

In this part we make a comparison between the different detectors results in terms of both accuracy and speed represented by mean average precision (mAP) and Frame Per Second (FPS) respectively.

For this purpose, the results achieved by these detectors on Pascal VOC 2007 and COCO datasets, was collected from different papers of each model ([29], [30], [32], [35], [33], [34], [37]) and we make them available in TABLE III and TABLE IV. We also present in TABLE II a list of GPUs used by each detector in its tests. Then, to analyze these results, we plot them together to get a full picture of variation in performance between the different detectors.

TABLE III show results on Pascal VOC 2007 The comparison of these methods as shown in Fig. 5 We note through the Fig. 5 an affinity at the accuracy level between deferent methods with a slight superiority of R-FCN by 80.5% mAP come after him YOLOv2 544 (544 for 544×544 input size) by 78.6% mAP. On the other hand, we notice the large difference in speed between the various methods. Tiny YOLO outperformed all other methods in terms of speed by 155 FPS. We also notice that YOLOv2 and SSD300 make a good

TABLE II. GPUs USED BY EACH MODEL

Detector	GPU
Fast R-CNN	Nvidia Tesla K40
Faster R-CNN	Nvidia Tesla K40
SSD	Nvidia Titan X
YOLO	Nvidia Titan X
YOLOv2	Geforce GTX Titan X
R-FCN	Nvidia Tesla K40
RetinaNet	Nvidia Tesla M40

compromise between speed and accuracy.

For the last couple years, many results are exclusively measured with the COCO object detection dataset. COCO dataset is harder for object detection and usually detectors achieve much lower mAP. TABLE IV show results on COCO dataset the comparison of these methods as shown in Fig. 6. We note through the Fig. 6 RetinaNet-100-800 achieved the best result in accuracy by 37.8 mAP followed by Faster RCNN-ResNet (use ResNet as backbone) wich achieved 34.9 mAP. YOLOv2 achieve the best performance in speed by 21.6 FPS.

Larger input size leads to better results in accuracy but it is the opposite of speed. The possibility of run a detector at different resolutions allowed an easy trade-off between speed

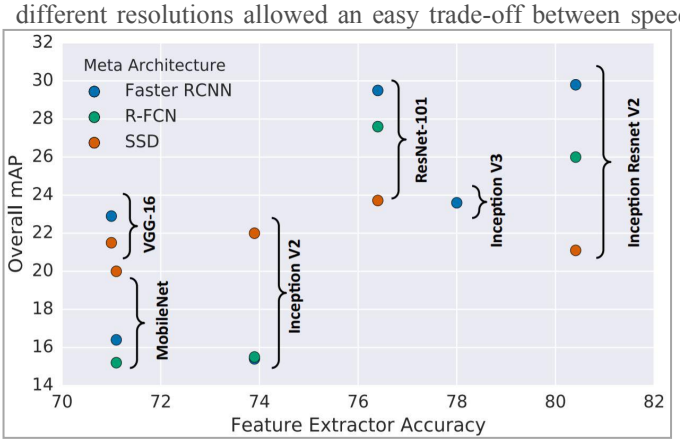

Fig. 4. The relation between accuracy of detector (measured by mAP on COCO) and accuracy of feature extractor (measured by top-1 accuracy on ImageNet).

and accuracy. We would also like to emphasize here that the choice of the feature extractors used to build our detector impacts detection accuracy. To clarify this relation between feature extractors performance and detection performance, Jonathan et al. studied in their paper [38] the relation between overall mAP of different Meta-Architectures of object detectors (Faster R-CNN , R-FCN and SSD) and the Top-1 Imagenet classification accuracy attained by the pre-trained feature extractor (VGG-16, MobileNet, Inception v2, ResNet-101, Inception v3, inception ResNet v2) used in each Meta-Architecture. The results is shown in Fig. 4 [38].

Fig. 4 indicates that the feature extractor classification accuracy has a significant influence on Faster R-CNN and R-FCN, while the performance of SSD is less influenced by its

Method	mAP	FPS
Tiny YOLO	52,7	155
YOLO	63,4	45
YOLO v2 288	69	91
YOLO v2 544	78,6	40
Fast R-CNN	70	0.5
Faster R-CNN	73,2	7
SSD 300	74,3	58
SSD 512	76,8	23
R-FCN	80.5	6

TABLE III. PASCAL VOC 2007 DATADET RESULTS

Method	mAP	FPS
YOLOv2	21,6	40
SSD321	28	16
R-FCN	29,9	12
SSD513	31,2	8
RetinaNet-50-500	32,5	14
RetinaNet-100-800	37,8	5
Faster RCNN	21,9	/
Faster RCNN(ResNet)	34,9	/

fea ture extractor classification accuracy. Also, SSD unable to take advantage of the power of a better feature extractor like ResNet and Inception unlike to Faster R-CNN and R-FCN, but, at the same time, is not much affected by using cheaper feature extractors.

v. Conclusion

In this paper we presented an overview of object detection methods based on deep learning. We started by a brief history of Convolutional Neural Networks and reviewed most important object detection method that used CNN architecture. We selected most used state of the art methods to compare them on their performances. Choice of a right object detection method is crucial and depends on the problem you are trying to solve and the set-up. Object Detection is the backbone of many practical applications of computer vision such as

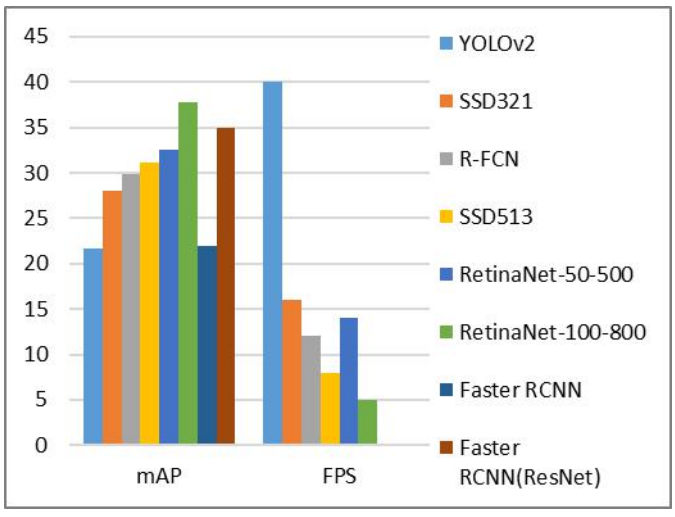

Fig. 6. Comparison of results achieved in COCO

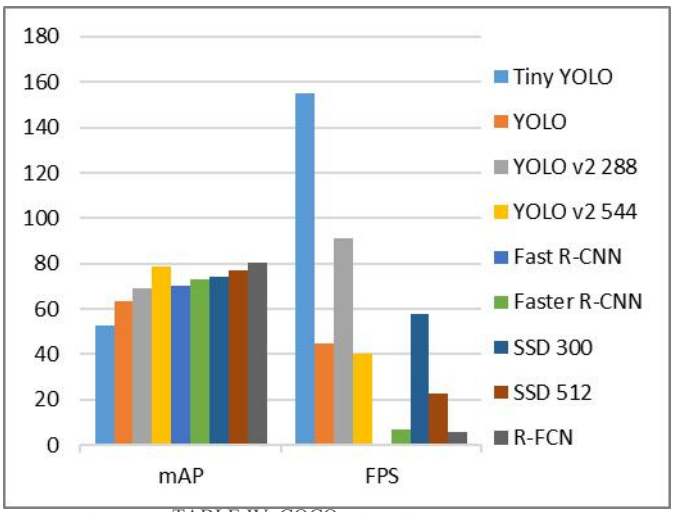

Fig. 5. Comparison of results achieved in PASCAL VOC 2007

autonomous cars, security and surveillance, and many industrial applications. Hopefully, this post gave you an intuition and understanding behind each of the popular algorithms for object detection.

References

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 32, pp. 1627-1645, 2010.

[2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001, pp. I-I.

[3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, 2005, pp. 886-893.

[4] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," The Journal of physiology, vol. 160, pp. 106-154, 1962.

[5] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in Competition and cooperation in neural nets, ed: Springer, 1982, pp. 267-285.

[6] R. Hetcht-Nielsen, "Theory of the backpropagation neural network," in Proceeding of International Joint Conference in Neural Networks, 1989, pp. 593-611.

[7] Y. LeCun, "Generalization and network design strategies," Connectionism in perspective, pp. 143-155, 1989.

[8] Y. LeCun, C. Cortes, and C. Burges. (01 january 2019). THE MNIST DATABASE of handwritten digits. Available: http://yann.lecun.com/exdb/mnist/

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, 2009, pp. 248-255.

[10] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," Computer vision and Image understanding, vol. 106, pp. 59-70, 2007.

[11] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," 2007.

[12] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, et al., "Microsoft coco: Common objects in context," in European conference on computer vision, 2014, pp. 740-755.

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," International journal of computer vision, vol. 88, pp. 303-338, 2010.

[14] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Citeseer2009.

[15] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in null, 2006, pp. 2169-2178.

[16] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large data set for nonparametric object and scene recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 30, pp. 1958-1970, 2008.

[17] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, 2010, pp. 3485-3492.

[18] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, et al., "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," arXiv preprint arXiv:1811.00982, 2018.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105.

[20] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, et al., "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, pp. 211-252, 2015.

[21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in European conference on computer vision, 2014, pp. 818-833.

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

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[25] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv preprint arXiv:1312.6229, 2013.

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

[27] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," International journal of computer vision, vol. 104, pp. 154-171, 2013.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE transactions on pattern analysis and machine intelligence, vol. 37, pp. 1904-1916, 2015.

[29] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440-1448.

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

[31] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in European conference on computer vision, 2014, pp. 391-405.

[32] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.

[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, et al., "Ssd: Single shot multibox detector," in European conference on computer vision, 2016, pp. 21-37.

[34] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in Advances in neural information processing systems, 2016, pp. 379-387.

[35] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517-6525.

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

[37] 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, 2018.

[38] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in IEEE CVPR, 2017.