iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 7 **Issue:** IX **Month of publication:** September 2019

DOI: http://doi.org/10.22214/ijraset.2019.9165

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.177
Volume 7 Issue IX, Sep 2019- Available at www.ijraset.com

A Review of Deep Learning based Object Detection Techniques

Jash Sheth[1], Sohan Shirodkar[2], Vishesh Vohra[3], Prof. Pankaj Sonawane[4]
[1, 2, 3]*B.E. Students,* [4]*Asst. Professor, Department of Computer Engineering, D. J. Sanghvi College of Engineering, Mumbai, Maharashtra, India.*

Abstract: ***The aim of object detection is to recognize instances of semantic objects belonging to a certain class within an image, accurately predict the location of the object in the image, and then to classify it according to a corresponding class label. In the past few years, there have been a lot of new and constantly improving models proposed for this task. Deep Learning based approaches, especially those involving Deep Convolutional Neural Networks, have been the most popular for good reason. In this paper, we aim to review the latest approaches in tackling the problem of object detection, while understanding the drawbacks of each approach as well as the improvements observed with the subsequent models. We then compare the results obtained by each model on popular datasets. In the last part, we aim to offer ideas for future work, scope for improvement and potential application areas.***

Keywords: Object Detection, Deep Learning, Convolution Neural Networks, Computer Vision, Object Recognition

I. INTRODUCTION

A fundamental problem of computer vision, Object detection [1] involves categorizing various components of an image into their corresponding classes. Object detection tasks can be either generic or specific to a problem area - such as Face Detection [2], Pedestrian Detection [3], Disease Identification, etc. As a result of the goal to build more robust models compatible in different applications, generic object detection has been gaining interest. Object detection has a variety of applications, including autonomous driving [4], security surveillance monitoring [5], captioning of images [6], robot vision and in the military as well. Owing to the vast potential of application areas, there has been tremendous research in the field of Object Detection in the past few decades.

A. Early Work

Traditional object detection approaches used handcrafted features to great effect. Early proposals in the field of object detection mainly included the Viola Jones detector [7], HOG detector [8] and Deformable Part-based Model (DPM) [9], among others. DPM performed particularly well, winning VOC challenges in 2007,08 and 09 on the PASCAL VOC dataset [10]. Despite the remarkable performance of early models, research work in the field of object detection stagnated after 2010, with most of the proposals involving a few tweaks to older architectures or ensembles involving previously successful models. Traditional approaches had a few notable flaws, this being highlighted by the lack of significant improvements during that time.

B. Use of Deep Learning

In 2012, Deep Convolutional Neural Networks (DCNNs) [11] started gaining a lot of research interest. DCNNs started finding applications in a lot of fields of computer vision, most prominently in Image Classification [12]. Researchers started exploring the potential of applying deep learning to object detection as well, with the introduction of the Overfeat network [13], which used a sliding window approach with a convolutional neural network. While this was a significant model in the evolution of object detection models, the breakthrough model R-CNN in 2014 [14], which outperformed the previous state-of-the-art by 30%, acted as the impetus for the immense interest in deep learning based object detection models in the following years. Since then, there have been a variety of models proposed, each suggesting unique ideas and improvements.

Earlier, there was a lack of high performance GPUs available for training, as well as insufficient training data for training powerful deep learning models. However, recent improvements in these aspects have empowered deep learning based models to comfortably surpass the performance of other models on popular datasets. In addition, these models are much faster. Earlier models took quite a bit of computation time, even the earliest deep learning approaches took approximately 47 seconds to process an image. However, the most recent approaches are fully capable of being used in real time, with the computation taking time in the order of milliseconds. Deep learning models have also outperformed other models in terms of mAP (mean average precision) scores.

C. Two Families of Object Detectors

Recently, there have been two approaches observed in the deep learning based object detectors: i) One stage approach, ii) Two stage approach. Two stage approach involves first generating region proposals from components of the input image, and then classifying into the corresponding class, generally involving an ROI pooling layer in between. Such object detectors have higher accuracy as compared to their one stage counterparts. Examples of these include R-CNN [14], SPPNet [15], Fast R-CNN [16], Faster R-CNN [17] and Mask R-CNN [18]. Object detectors following the one stage approach directly make predictions in one step following a unified framework. Such object detectors are much faster in speed, making them suitable for real time applications. Few of the most widely used one stage object detectors are YOLO [19], RetinaNet [20], SSD [21], RefineDet [22], and YOLOv2 [23].

II. LITERATURE SURVEY

In this section we review the most popular one stage and two stage object detection models, as well as their improvements and limitations.

A. Two Stages Approaches

1) *R-CNN:* Proposed by Ross Girshick et al. [14], in 2014, R-CNN introduced the concept of a CNN based two-stage object detector. R-CNN uses Selective Search [24] along with the AlexNet [25]. The R-CNN pipeline consists of 3 major constituents: (i) Region proposal generation, (ii) Feature extraction using CNN and (iii) Region classification. Selective search is used to generate 2000 region proposals from the input image, which are then resized into a fixed resolution and fed into a CNN to extract a fixed length (e.g. 4096 dimensional) feature vector from each region proposal. Linear SVM [26] classifiers are used to classify the object into the corresponding category.

Although R-CNN managed to perform significantly better than the previous state-of-the-art methods, it had a few major drawbacks. (i) Aspect ratio and size of the image got compromised during transformation of the region proposal into a fixed resolution. (ii) Since there are 2k region proposals, training occupied a lot of space on the disk, as well as took a lot of time. (iii) R-CNN followed a multistage pipeline, which is difficult to optimize. (iv) R-CNN is too slow to be implemented in real time.

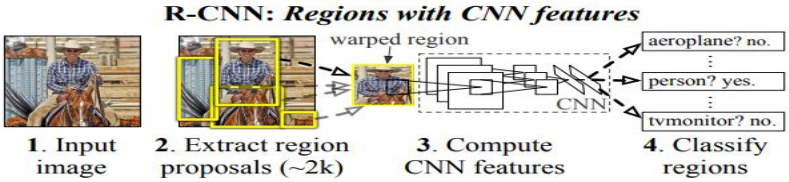


Fig. 1 R-CNN

2) *SPP-Net:* Spatial Pyramid Pooling Network (SPP-Net) was proposed by K. He et al. [15] in 2014, aiming to improve the speed of the R-CNN architecture. Unlike Alex Net in R-CNN, the network in SPP-Net did not require a fixed size input region proposal. In fact, the need to crop or scale every input image, as well as the problem of image distortion was solved by adding a Spatial Pyramid Pooling (SPP) [27] layer after the last convolutional layer, just before the fully connected (FC) layers of R-CNN. This layer re-used the feature maps of the last conv layer to generate fixed length output vectors. Therefore, for region proposals of any size or aspect ratio, SPP generates a fixed size output representation.

SPP-Net takes significantly less time as compared to R-CNN, proving advantageous in this aspect. However, it still suffers a few crucial drawbacks. (i) SPP-Net, like R-CNN follows a multistage pipeline, thereby making it harder to optimize, as well as using a lot of storage space. (ii) All parameters up till the SPP layer were not tuned, thereby causing the parameters to remain constant. Only the FC layers were fine tuned in the back propagation process, resulting in lower accuracy for very deep networks.

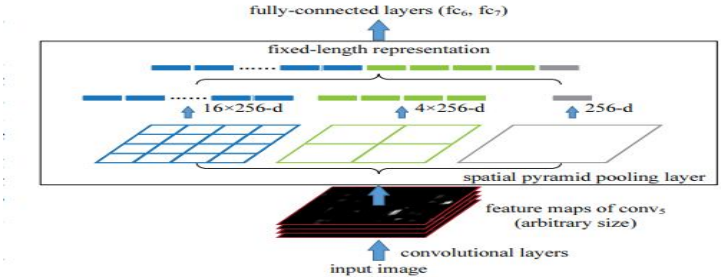


Fig. 2 SPP-Net

3) *Fast R-CNN:* Soon after R-CNN, the same author, Girshick introduced Fast R-CNN [16], a model that combined the advantages of both R-CNN and SPP-Net, as well as improved on a few of their limitations. Unlike R-CNN, Fast R-CNN allows shared computation among region proposals by generating a feature map from the whole input image by passing it through the conv layers. Region features are extracted using a special ROI Pooling layer, a special case of the Spatial Pyramid Pooling layer that was proposed in SPP-Net. The fixed length output vectors generated by the ROI Pooling layer are then fed into the FC layer. The vectors are fed into a classification layer for generating SoftMax outputs and bounding box regression layers. Using a single step training process for all the layers, Fast R-CNN is significantly faster than R-CNN (reducing training time from 84 to 9 hours) and SPP-Net in training and testing, as well as better in detection accuracy. In addition, Fast R-CNN requires lesser storage space as well.

Despite the massive improvement in performance that Fast R-CNN has over previously proposed architectures, it still relies on traditional techniques like Selective Search in order to generate region proposals. This method is slow and computationally demanding, making it impractical to apply real time systems.

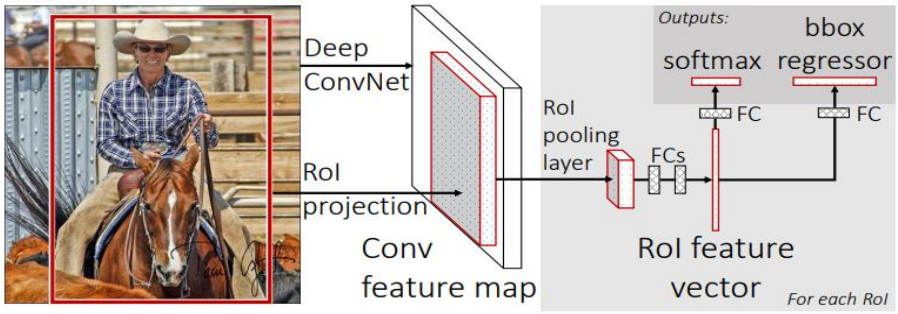


Fig. 3 Fast R-CNN

4) *Faster R-CNN:* The need for a faster, more optimized approach to improve the Fast R-CNN led to the introduction of Faster R-CNN [17], proposed by Ren et al. in 2015. Faster R-CNN attempted to solve the problem of high computational power required by the selective search algorithm, by making use of Region Proposal Network (RPN), a fully convolutional network [28]] used to generate a set of region proposals from an input image. This led to an end-to-end framework being formed when integrated with the Fast R-CNN network. RPN is used for generating region proposals, whereas the same Fast R-CNN backbone was used for region classification.

During region classification, each fixed size feature vector undergoes the FC layers individually during Faster R-CNN, a shortcoming that may make the overall computation very slow for large number of region proposals.

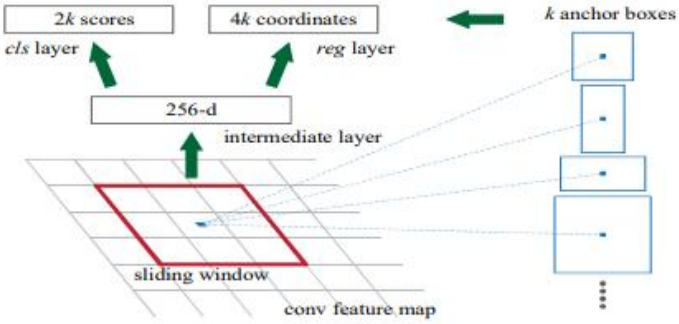


Fig. 4 Faster R-CNN

5) *Mask R-CNN:* The need for a faster, more optimized approach to improve the Fast R-CNN led to the introduction of Faster R-CNN [17], proposed by Ren et al. in 2015. Faster R-CNN attempted to solve the problem of high computational power required by the selective search algorithm, by making use of Region Proposal Network (RPN), a fully convolutional network [28]] used to generate a set of region proposals from an input image. This led to an end-to-end framework being formed when integrated with the Fast R-CNN network. RPN is used for generating region proposals, whereas the same Fast R-CNN backbone was used for region classification.

During region classification, each fixed size feature vector undergoes the FC layers individually during Faster R-CNN, a shortcoming that may make the overall computation very slow for large number of region proposals.

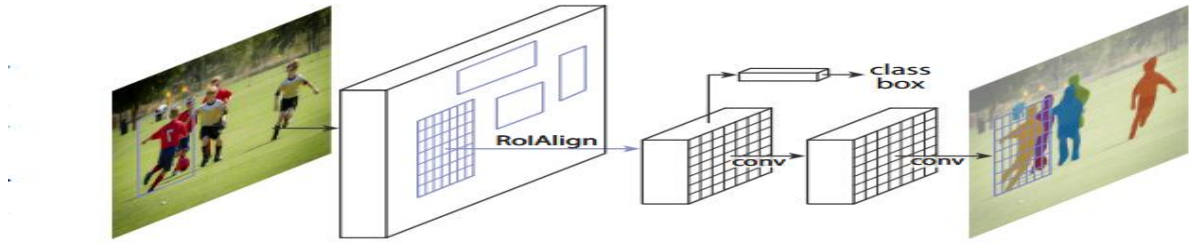


Fig. 5 Mask R-CNN

6) *R-FCN:* Region based Fully Convolutional Network (R-FCN), proposed by Dai et al. [29], aims to resolve the primary shortcoming of Faster R-CNN and Fast R-CNN by proposing a network that is fully convoluted, with the absence of any FC layers. By this approach, almost all the computations are shared throughout the whole image. R-FCN makes an improvement over Faster R-CNN mainly in the ROI layer. One of the main advantages of R-FCN is that the computation time required for a convolution layer is faster than that of a fully connected layer. R-FCN also avoids the computationally expensive process of cropping or resizing an image, by allowing images of various dimensions to be fed into the network.

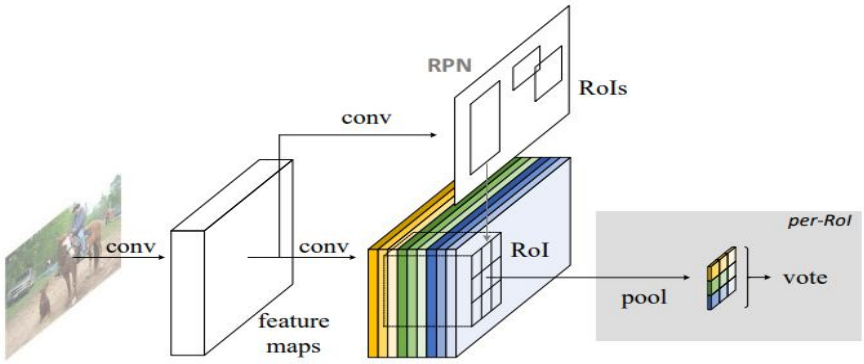


Fig. 6 R-FCN

B. One Stage Approaches

1) *YOLO:* Under You Only Look Once (YOLO) [19], object detection is considered to be a regression task and instead of pipelines a single convolutional network is used that predicts as well as classifies bounding boxes. An image is divided into a grid. Each grid cell predicts a certain number of bounding boxes along with predictions - x, y, w, h and confidence. (x, y) represent the center of the box with respect to the grid cell that contains it. Symbols w and h represent the width and height relative to the whole image. The confidence score is the likelihood of the object prediction being accurate. Since each grid cell can have only 2 bounding boxes, the model cannot classify small objects that occur in groups successfully. In comparison to other models YOLO uses coarser features due to the down sampling layers present. It also faces difficulty when dealing with objects in unusual aspect ratios. YOLO is much faster than the R-CNN models but not as accurate.

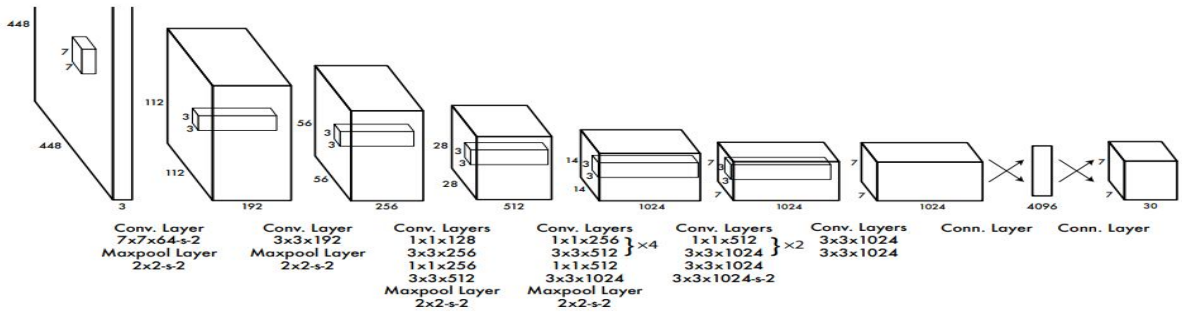


Fig. 7 YOLO

2) *SSD: Single Shot Multibox Detector (*SSD) [21] strives to obtain a balance between speed and accuracy. It was designed to implement object detection in real time. It's faster than Faster R-CNN but not as accurate while it's more accurate than YOLO but not as fast. In YOLO, the aspect ratios of bounding boxes are fixed, while SSD uses anchor boxes of different aspect ratios. This technique is similar to that of Faster R-CNN. Instead of using a region proposal network small convolution filters are used to compute object location and class. For large objects the accuracies of SSD and Faster R-CNN are comparable but as the object size decreases SSD's accuracy drops significantly.

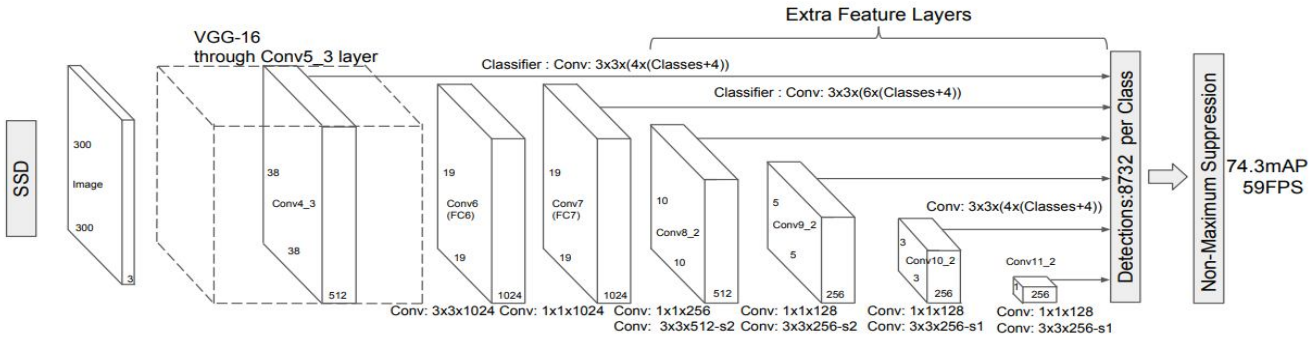


Fig. 8 SSD

3) *YOLO9000:* YOLO9000 [23] utilizes the base of Yolov2 but with 3 anchor boxes rather than 5. It is trained over the combined WordTree structure acquired subsequent to merging the classification and detection datasets. The COCO detection dataset [31] and 9000 classes from ImageNet [32] are merged. Since, the number of labels in ImageNet is more than that in the COCO dataset, the COCO dataset is oversampled to create a balance. Only the classification loss is back propagated by finding the bounding box that predicts the highest probability for that class and computing the loss on that predicted tree.

4) *RetinaNet:* RetinaNet uses feature pyramid networks and the focal loss function to improve performance in comparison to other single stage object detection models like SSD and YOLO. It matched the speed of one-stage detectors and at the same time surpassed the accuracy of two-stage detectors. Using focal loss for object detection was proposed by Ross Girshick et al. [20,30], to reduce class imbalance and thus improve performance. Feature pyramids were generally avoided due to requiring large computations. Ross Girshick proposed constructing feature pyramids at a marginal cost using deep convolutional networks. The architecture built on top of a feature pyramid is called a Feature Pyramid Network. Implementing a feature pyramid network in a basic Faster R-CNN model achieves state of the art results.

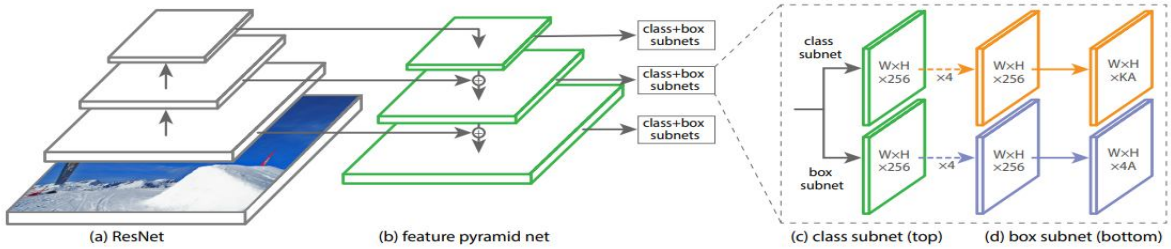


Fig. 9 RetinaNet

5) *RefineDet:* RefineDet [22] is a single-shot-based object detection algorithm. It achieves better accuracy than two-stage methods such as R-CNN and R-FCN while also offering relatively efficiency close to one stage object detectors such as YOLOv2 and SSD. RefineDet consists of two interconnected modules - the anchor refinement module and the object detection module. This improves the architecture of the one-stage method to overcome the class imbalance problem and improve detection accuracy. RefineDet produces 80% mAP (mean Average Precision, a popular metric to measure the accuracy of object detectors). Achieving more than 80% accuracy on the PASCAL VOC 2007 dataset while also being able to support real time implementation is a feat first achieved by RefineDet. RefineDet achieves top accuracy with high efficiency, mainly thanks to the design of two interconnected modules.

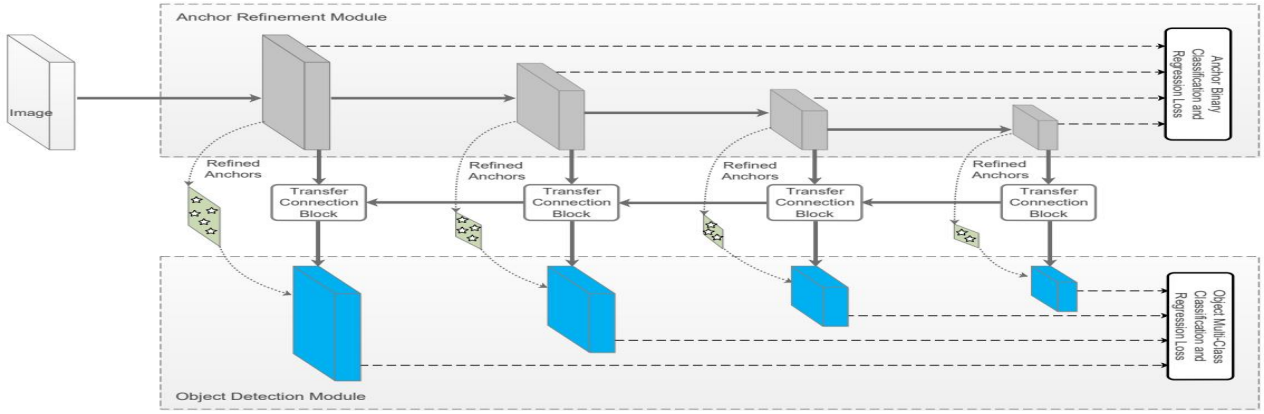


Fig. 10 RefineDet

III. SUMMARY OF VARIOUS MODELS

Table I offers a general summary on the various models reviewed in this paper. It includes key aspects of difference among the models like the technique used for generating region proposals and the number of stages the model performs in, among other factors. One stage object detectors tend to be faster, thereby most of have been considered for real time applications.

The table illustrates the key features that the models have as the years progress, as most of the proposed models after the introduction of Faster R-CNN have been able to be trained end-to-end. The neural networks that have been used as a backbone network for each model can also be seen, highlighting the impact that these prominent D-CNNs have had.

Table I
Summary Of Various Models

Model	No of stages	Backbone D-CNN	Allows multi-scale input	Region proposal generation	End to end training	Softmax layer	Real time speed	Year
R-CNN	2	AlexNet	No	Selective search	No	Yes	No	2014
SPP-Net	2	ZFNet	Yes	Edge Boxes	No	Yes	No	2014
Fast R-CNN	2	AlexNet VGGM VGG16	Yes	Selective search	No	Yes	No	2015
Faster R-CNN	2	ZFNet VGG	Yes	RPN	Yes	Yes	No	2015
R-FCN	2	ResNet 101	Yes	RPN	Yes	No	No	2016
Mask R-CNN	2	ResNet 101 ResNeXt 101	Yes	RPN	Yes	Yes	No	2017
YOLO	1	GoogLeNet like	No	-	Yes	Yes	Yes	2014
SSD	1	DarkNet	No	-	Yes	No	No	2016
YOLOv2	1	VGG16	No	-	Yes	Yes	Yes	2017
RetinaNet	1	ResNet 101	Yes	FPN	Yes	Yes	Yes	2017
RefineDet	1	VGG16	Yes	ARM	Yes	Yes	Yes	2017

IV. PERFORMANCE ANALYSIS

It is important to contrast the performance of the most prominent models on popular datasets like COCO and PASCAL VOC in order to have a better idea about how useful each model can be in a subsequent application. Datasets keep updating over the years, with new training examples or features being added. A model tested on an older version of a dataset like PASCAL VOC 2007, if compared against a model tested on a newer dataset may not offer a fair comparison of their performance.

It is clearly observed that models following a two stage approach generally have higher mean average precision (mAP) scores, indicating better accuracy. This is due to the fact that object detectors following one stage approach prioritize faster speed over better accuracy, as seen by their lower mAP scores and real time speed. RetinaNet, which combines the advantages of both families of object detectors, is seen to have higher accuracy than two stage object detectors as well as faster speed than one stage object detectors.

Table II
Summary of Various Models

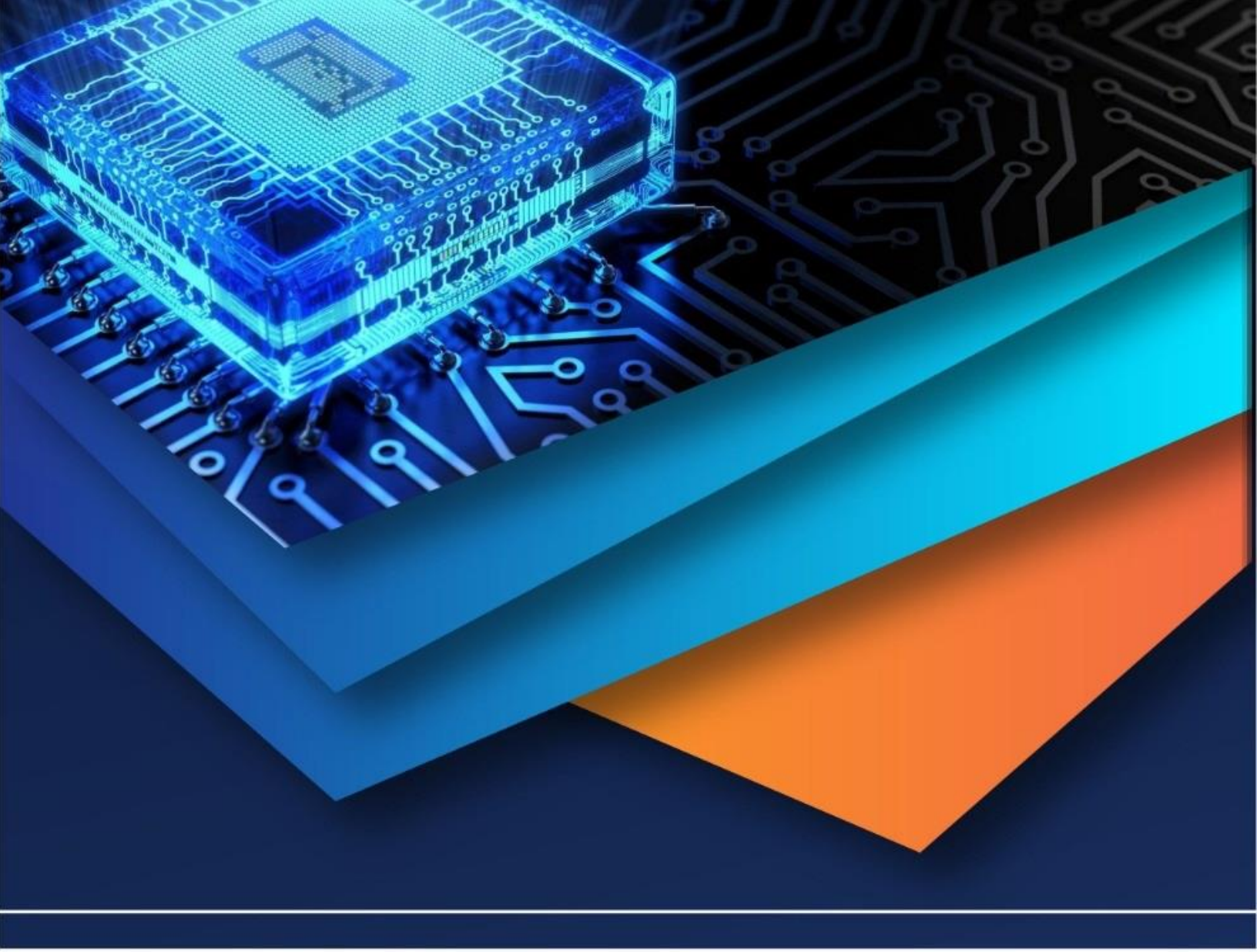
Model	PASCAL VOC 2007	PASCAL VOC 2010	PASCAL 2012	COCO 2015 (IoU =0.5)	COCO 2015 (IoU = 0.75)	COCO 2015 (Official metric)	COCO 2016 (IoU = 0.5)	COCO 2016 (IoU = 0.75)	COCO 2016 (Official metric)	Real time speed	Stage
R-CNN		62.4%								No	Two
SPPNet										No	Two
Fast R-CNN	70.0%	68.8%	68.4%							No	Two
Faster R-CNN	78.8%		75.9%							No	Two
R-FCN	82.0%			53.2%		31.5%				No	Two
Mask R-CNN							62.3%	43.3%	39.8%	No	Two
YOLO	63.7%	57.9%								Yes	One
SSD	83.2%		82.2%	48.5%	30.3%	31.5%				No	One
YOLOv2	78.6%			44.0%	19.2%	21.6%				Yes	One
RetinaNet						37.8%				Yes	One
RefineDet						41.8%				Yes	One

V. CONCLUSION

Object detection has been widely researched in recent years, with the state-of-the-art being constantly beaten by improved models. In this paper, we reviewed the most widely cited and most influential ones among the many proposed object detectors in recent years. We offer insights into the architecture, working and the key differences between the models while also stating their improvements and shortcomings. Finally, we summarized the key aspects of the models we reviewed, along with their performance on popular datasets. With the help of this paper, we aim to help researchers understand the recent trends in the field of object detection as well as understand the various deep learning based object detectors that have been proposed in the past decade. We hope this paper can prove to be helpful for encouraging faster and more robust deep learning based object detectors in the future.

REFERENCES

[1] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no 9, p. 1627, 2010.
[2] K. K. Sung and T. Poggio, "Example-based learning for view-based human face detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 20, no. 1, pp. 39–51, 2002.
[3] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, S. Yan, Scale-aware fast r-cnn for pedestrian detection, in: IEEE Transactions on Multimedia, 2018.
[4] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving" in ICCV, 2015
[5] J.C. Nascimento, J.S. Marques, "Performance evaluation of object detection algorithms for video surveillance" in IEEE Transactions on Multimedia, vol. 8 issue 32, 2006.
[6] Justin Johnson, Andrej Karpathy, Li Fei-Fei; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4565-4574.
[7] P. Viola and M. Jones, "Robust real-time face detection," Int. J. of Comput. Vision, vol. 57, no. 2, pp. 137–154, 2004.
[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in CVPR, 2005.
[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, pp. 1627–1645, 2010.
[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," International Journal of Computer Vision, vol. 88, no. 2, pp. 303–338, Jun 2010.
[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in NIPS, 2012.
[12] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in ACM MM, 2014
[13] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv:1312.6229, 2013.
[14] R. B. Girshick, "Fast R-CNN," CoRR, vol. abs/1504.08083, 2015.
[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," CoRR, vol. abs/1406.4729, 2014.
[16] R. B. Girshick, "Fast R-CNN," CoRR, vol. abs/1504.08083, 2015.
[17] S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun, "Object detection networks on convolutional feature maps," CoRR, vol. abs/1504.06066, 2015.
[18] K. He, G. Gkioxari, P. Dollar, and R. B. Girshick, "Mask r-cnn", 2017 IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988, 2017.
[19] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," CoRR, vol. abs/1506.02640, 2015.
[20] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection", CoRR, vol. abs/1708.02002, 2017.
[21] 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.
[22] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," CoRR, vol. abs/1711.06897, 2017
[23] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525, 2017.
[24] J.R. Uijlings, K. E. Van De Sande, T. Gevers, A. W. Smeulders, Selective search for object recognition, in: IJCV, 2013.
[25] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: NeurIPS, 2012.
[26] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, B. Scholkopf, Support vector machines, in: IEEE Intelligent Systems and their applications, 1998.
[27] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in CVPR, 2006.
[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," CoRR, vol. abs/1411.4038, 2014.
[29] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, in: NeurIPS, 2016.
[30] T.-Y. Lin, P. Dollar, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in CVPR, 2017.
[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in ECCV, 2014.
[32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in CVPR, 2009.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY