# Deep Learning based Neck Models for Object Detection: A Review and a Benchmarking Study

Sara Bouraya, Abdessamad Belangour
Laboratory of Information Technology and Modeling
Hassan II University, Faculty of Sciences Ben M'sik
Casablanca, Morocco

***Abstract*—Artificial intelligence is the science of enabling computers to act without being further programmed. Particularly, computer vision is one of its innovative fields that manages how computers acquire comprehension from videos and images. In the previous decades, computer vision has been involved in many fields such as self-driving cars, efficient information retrieval, effective surveillance, and a better understanding of human behaviour. Based on deep neural networks, object detection is actively growing for pushing the limits of detection accuracy and speed. Object Detection aims to locate each object instance and assign a class to it in an image or a video sequence. Object detectors are usually provided with a backbone network designed for feature extractors, a neck model for feature aggregation, and finally a head for prediction. Neck models, which are the purpose of study in this paper, are neural networks used to make a fusion between high-level features and low-level features and are known by their efficiency in object detection. The aim of this study to present a review of neck models together before making a benchmarking that would help researchers and scientists use it as a guideline for their works.**

***Keywords*—Object detection; deep learning; computer vision; neck models; feature aggregation; feature fusion**

## I. Introduction

Object detection is often called image detection, object identification, and object recognition; and all these concepts are synonymous. It is a computer vision method for locating instances of objects in an image or video sequence. Object detection algorithms, therefore, typically benefit from machine learning techniques or deep learning techniques to gain meaningful results. When humans look at images or videos, they could locate and recognize objects of interest easily. The goal of object detection is to mimic this intelligence using a computer. With recent advancements in Deep Learning-based computer vision models, Object Detection use cases are spreading more than ever before. A wide range of applications is implemented, for instance, self-driving cars, object tracking, anomaly detection, and video surveillance.

Object Detection could be divided into two main categories Deep Learning-based techniques and Machine Learning based techniques. Deep Learning based techniques could be separated into two approaches one stage detectors and two-stage detectors. Object Detection based Deep Learning approaches are a set of models of Deep Learning, starting from input, then a backbone for feature extraction model, then neck model for feature fusion, and finally a head model class/box network.

The neck of the object detector refers to the additional layers existing between the backbone [1] and the head. Their role is to collect feature maps from different stages. The neck models are composed of several top-down paths and several bottom-up paths. The idea behind this feature aggregation existing in this model is to allow low-level features to interact more directly with high-level features, by mixing information from this high-level feature with the low-level feature. They reach aggregation and feature interaction across many layers, since the distance between the two feature maps is large. Several methods can reach be implemented in this part, for example, PAN [2] or FPN [3] (see Fig. 1).

Head is the last model of object detection, predicts bounding boxes and classes of objects and could be a sparse prediction that belongs to One-stage detectors such as YOLO [4] , SDD [5], CenterNet [6], or a Dense prediction that belongs to Two-stage detectors, such as Fast R-CNN [7], Faster R-CNN [8], Mask R-CNN [9] (see Fig. 1). On the one hand, One Stage detectors have high inference speeds, these models predict bounding boxes in a one or single step without using region proposals. On the other hand, two stage detectors have high localization and recognition accuracy. Firstly, they use a Region Proposal Network to generate regions of interests; secondly, they send the region proposals for object classification and bounding-box regression.

We aim that our benchmarking study can provide a timely comparison of neck models of object detection for practitioners and researchers to further master research on object detection models. The rest of our study is organized as follows: In Section 2, we are going to discuss the different existing related works about feature aggregation. In Section 3, we list the neck neural networks about object detection used for feature fusion, their architecture is discussed also in their categories. In Section 4, our comparative study is presented. In Section 5, we highlight the different recognizable results and Section 6 covers the discussion. Finally, in Section 7, we conclude and discuss future directions.

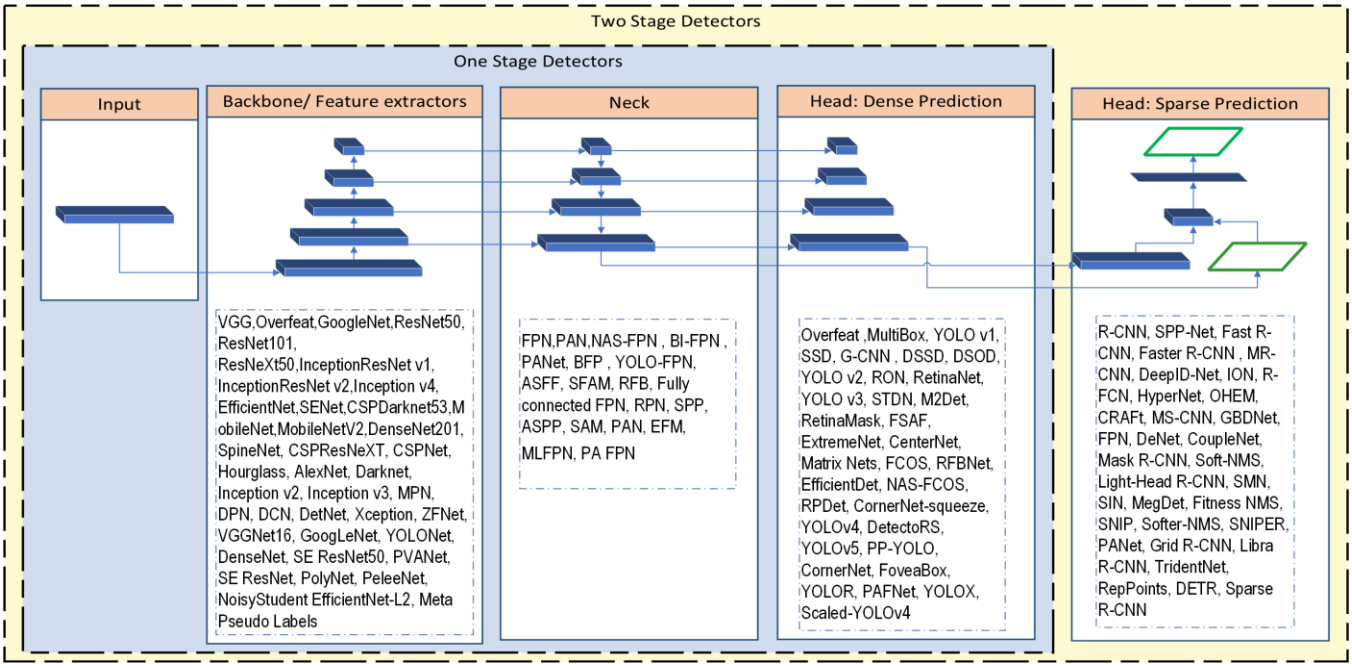


Fig. 1. Models' Taxonomy of Object Detectors in each Part Backbone, Head, and Neck.

## II. Related Work

Several scientific works and researches have been implemented to develop and evolve Object Detection applications and systems and depend on enormous methodologies of the deep learning era, machine learning era and other eras. Several researchers and scientists are expanding their implementation and research to develop and apply enormous methodologies. Such us the case of feature aggregation methods that are used to make a connection between low and high feature for better object recognition in video sequence and images. Feature aggregation is used widely in action recognition [10], [11], [12], [13], [14] and video description [15],[16]. Most of these methods use recurrent neural network (RNNs) in order to aggregate features from consecutive frames on the one hand. Exhaustive temporal-spatial convolution is used to extract temporal-spatial features, on the other hand. U-Net [17] was proposes to concatenate features from low level to high-level for medical image segmentation, and it achieved great success in that field. In order to gain an outstanding feature for object detection, the FPN stands for Feature Pyramid Networks aggregated both the transformed feature from the bottom-up weighted pyramid and the top-down lateral convolutions through a simple sum operation. Relied on Feature Pyramid Networks, several extensive works [18], [19], [20], [2] define new option on connectivity between scales. Attention based models also prove their efficiency in several applications of deep learning era [21], [22], [23], [24], [25], [26]. Self-attention models by measuring and applying a context relied encoding summarized from a dimension of feature. All these works cited propose to aggregate and fuse features via element-wise concatenation or summation.

## III. Background

Since Feature Pyramid Networks appearance, the focus of this work is the object detector neck, the existing part between the backbone and the head. These techniques are useful for many reasons.

*1) Aggregation network models (FPN)*: FPN [3] is a top-down architecture with lateral connections, it is implemented in building high-level semantic feature maps at all scales (see Fig. 2).

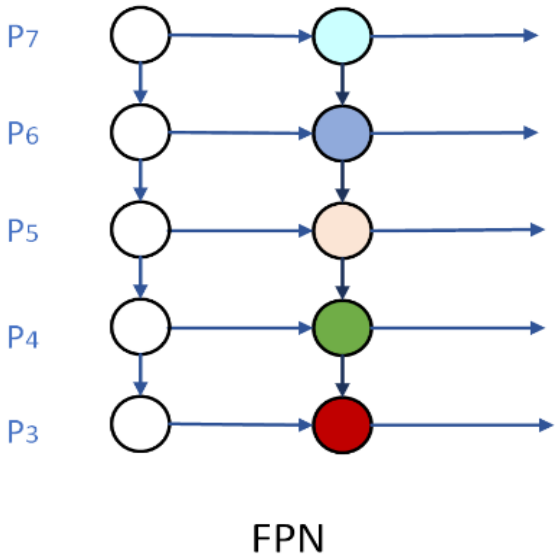


Fig. 2. FPN Architecture.

*2) Neural architecture search FPN (NAS-FPN)*: NAS-FPN [19] consists of a combination of top-down and bottom-up connections to fuse features across scales (see Fig. 3).

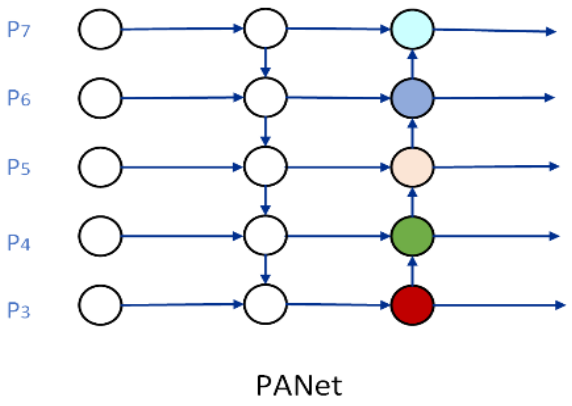


Fig. 3. PANet Architecture.

*3) Neural architecture search FPN (NAS-FPN)*: NAS-FPN [19] consists of a combination of top-down and bottom-up connections to fuse features across scales (see Fig. 4).

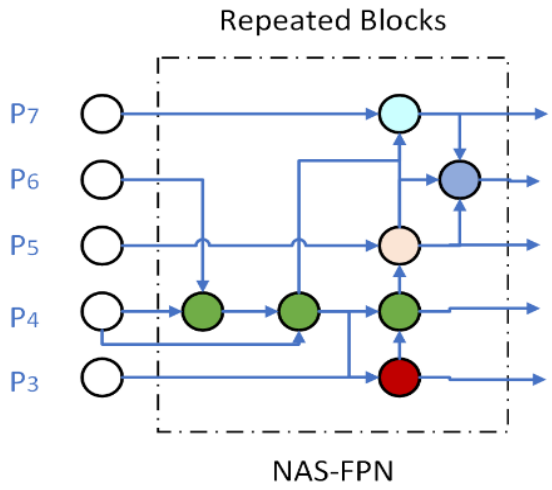


Fig. 4. NAS-FPN Architecture.

*4) Bi-directional feature pyramid network (BiFPN)*: BiFPN [27] is a type of feature pyramid network that allows fast and easy multi-scale feature fusion. BiFPN incorporates the other feature fusion models. It enables information to flow in the top-down and bottom-up directions, while using efficient and regular connections. This network improves the connections by removing some nodes and treats each bidirectional path as a feature network layer (Fig. 5).

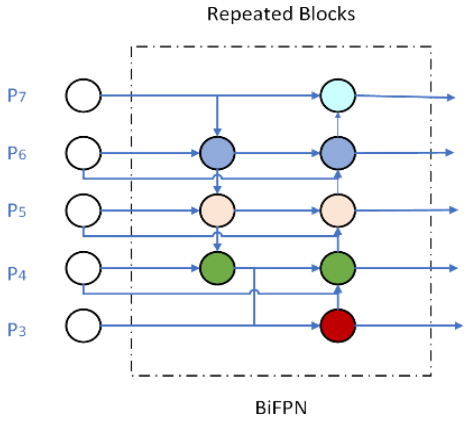


Fig. 5. BiFPN Architecture.

Based on the architecture above PANET is more performant then FPN et NAS-FPN, but the computation cost is higher.

*5) Fully-connected FPN*: Fully-connected, the calculation is the most complex all scales use the most complete connection (see Fig. 6).

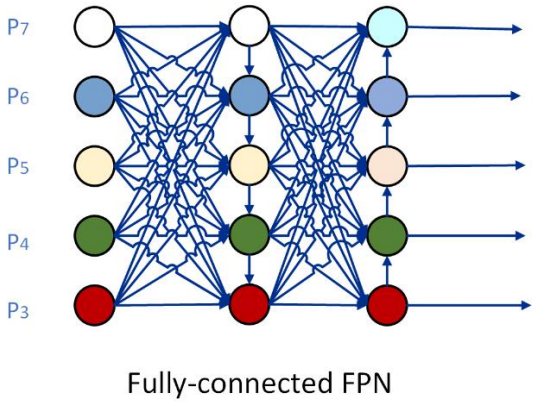


Fig. 6. Fully-Connected FPN Architecture.

*6) Simplified PANet*: Simplified PANet, this method simplifies and removes only one input node (see Fig. 7).

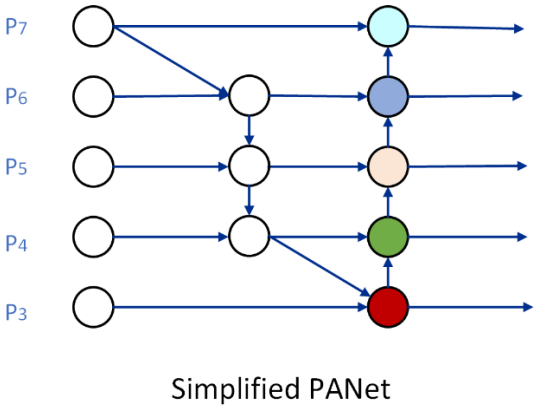


Fig. 7. Simplified FPN Architecture.

## IV. COMPARISON

Table I below illustrates the models that we are going to compare based on different comparison metrics. The measures are gathered carefully to cover several methods.

This table illustrates the deep learning models used for the object detection task of the COCO dataset. It defines the used models for the prediction for classification and bounding boxes. The Backbone determines the backbone used for feature extraction the number associated refers to the number of layers, and finally, the neck illustrates the feature aggregation network used.

Table I contains the model's name, Reference, Journal year, Year, Backbone, Neck, AP, AP50, AP75, APS, APM, APL (see Table I).

TABLE I. DETAILED COMPARISONS ON MULTIPLE POPULAR BASELINE OBJECT DETECTORS ON THE COCO DATASET

| Model Ref | Journal | Model | Backbone | Neck | AP | $AP_{50}$ | $AP_{75}$ | $AP_S$ | $AP_M$ | $AP_L$ |
|---|---|---|---|---|---|---|---|---|---|---|
| [18] | CVPR 2019 | Libra R-CNN | ResNet-50 | FPN | 38.7 | 59.9 | 42.0 | 22.5 | 41.1 | 48.7 |
| | | Libra R-CNN | ResNet-101 | FPN | 40.3 | 61.3 | 43.9 | 22.9 | 43.1 | 51.0 |
| | | Libra R-CNN | ResNeXt-101 | FPN | 43.0 | 64 | 47 | 25.3 | 45.6 | 54.6 |
| [8] | | Faster R-CNN | ResNet-50 | FPN | 37.8 | 58.7 | 40.6 | 21.3 | 41.0 | 49.5 |
| | | Faster R-CNN | ResNet-50 | AdaFPN | 39.0 | 58.8 | 41.8 | 22.6 | 42.3 | 50.0 |
| | | Faster R-CNN | ResNet-50 | AugFPN | 38.8 | 61.5 | 42.0 | 23.3 | 42.1 | 47.7 |
| | | Faster R-CNN | ResNet-101 | AugFPN | 41.5 | 63.9 | 45.1 | 23.8 | 44.7 | 52.8 |
| | | Faster R-CNN | ResNext-101- 32x4d | AugFPN | 41.9 | 64.4 | 45.6 | 25.2 | 45.4 | 52.6 |
| | | Faster R-CNN | ResNext-101-64x4d | AugFPN | 43.0 | 65.6 | 46.9 | 26.2 | 46.5 | 53.9 |
| | | Faster R-CNN | MobileNet-v2 | AugFPN | 34.2 | 56.6 | 36.2 | 19.6 | 36.4 | 43.1 |
| [28] | ICCV 2019 | FCOS | ResNet-50 | AugFPN | 37.9 | 58.0 | 40.4 | 21.2 | 40.5 | 47.9 |
| | | FCOS | ResNet-50 | FPN | 39.1 | 57.9 | 42.1 | 23.3 | 43.0 | 50.2 |
| | | FCOS | ResNet-50 | AdaFPN | 40.1 | 58.6 | 43.2 | 24.1 | 43.6 | 50.6 |
| | | FCOS | ResNeXt-101 | FPN | 42.7 | 62.2 | 46.1 | 26.0 | 45.6 | 52.6 |
| [9] | ICCV 2017 | Mask R-CNN | ResNet-101 | FPN | 38.2 | 60.3 | 41.7 | 20.1 | 41.1 | 50.2 |
| | | Mask R-CNN | ResNeXt-101 | FPN | 39.8 | 62.3 | 43.4 | 22.1 | 43.2 | 51.2 |
| | | Mask R-CNN | ResNet-50 | AugFPN | 39.5 | 61.8 | 42.9 | 23.4 | 42.7 | 49.1 |
| | | Mask R-CNN | ResNet-101 | AugFPN | 42.4 | 64.4 | 46.3 | 24.6 | 45.7 | 54.0 |
| | | Mask R-CNN | ResNet-50 | $A^2$ -FPN | 36.6 | 59.3 | 39.1 | 19.8 | 39.3 | 48.0 |
| | | Mask R-CNN | ResNet-101 | $A^2$ -FPN | 37.9 | 60.8 | 40.5 | 20.6 | 41.8 | 50.1 |
| [29] | CVPR 2018 | CascadeR-CNN | ResNet-50 | FPN | 36.5 | 59 | 39.2 | 20.3 | 38.8 | 46.4 |
| | | CascadeR-CNN | ResNet-101 | FPN | 38.8 | 61.1 | 41.9 | 21.3 | 41.8 | 49.8 |
| | | CascadeR-CNN | ResNet-101 | AC-FPN | 45.0 | 64.4 | 49.0 | 26.9 | 47.7 | 56.6 |
| [30] | ICCV 2017 | RetinaNet | ResNet-101 | FPN | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| | | RetinaNet | ResNeXt-101 | FPN | 40.8 | 61.1 | 44.1 | 24.1 | 44.2 | 51.2 |
| | | RetinaNet | ResNet-50 | AugFPN | 37.5 | 58.4 | 40.1 | 21.3 | 40.5 | 47.3 |
| | | RetinaNet | MobileNet-v2 | AugFPN | 34.0 | 54.0 | 36.0 | 18.6 | 36.0 | 44.0 |
| [31] | arXiv 2019 | RetinaMask | ResNet-50 | FPN | 39.4 | 58.6 | 42.3 | 21.9 | 42.0 | 51.0 |
| [32] | CVPR 2019 | Grid R-CNN | ResNeXt-101 | FPN | 43.2 | 63.0 | 46.6 | 25.1 | 46.5 | 55.2 |
| [33] | CVPR 2019 | HTC | ResNeXt-101 | FPN | 47.1 | 63.9 | 44.7 | 22.8 | 43.9 | 54.6 |
| | | HTC | ResNet-50 | FPN | 38.4 | 60.0 | 41.5 | 20.4 | 40.7 | 51.2 |
| | | HTC | ResNet-101 | FPN | 39.7 | 61.8 | 43.1 | 21.0 | 42.2 | 53.5 |
| | | HTC | ResNet-50 | A2 -FPN | 39.8 | 62.3 | 43.0 | 21.6 | 42.4 | 52.8 |
| | | HTC | ResNet-101 | A2 -FPN | 40.8 | 63.6 | 44.1 | 22.3 | 43.5 | 54.4 |
| | | HTC | ResNeXt -101 | A2 -FPN | 42.1 | 65.3 | 45.7 | 23.6 | 44.8 | 56.0 |
| [34] | CVPR 2020 | DetectRS | ResNeXt-101-DCN | RFP | 53.3 | 71.6 | 58.5 | 33.9 | 56.5 | 66.9 |
| [35] | arXiv 2021 | CenterNet2 | Res2Net-101-DCN | BiFPN | 56.4 | 74.0 | 61.6 | 38.7 | 59.7 | 68.6 |

Average Precision (AP)

AP % AP at IoU=.50:.05:.95

APIoU=.50 % AP at IoU=.50

APIoU=.75 % AP at IoU=.75

AP Across Scales:

APsmall % AP for small objects: area < 322

APmedium AP for medium objects: 322 < area < 962

APlarge AP for large objects: area >962

## V. Result

In this part, we are going to discuss the performance of different methods cited in Table I Libra R-CNN, Faster R-CNN, FCOS, Mask R-CNN, Cascade R-CNN, RetinaNet, RetinaMask, Grid R-CNN, HTC, DetectRS, CenterNet2 methods based on different feature aggregation networks and different backbone networks. In each model, we tried to fix either a backbone or a neck and see how the performance behave. These results show us the importance of both feature aggregation networks and feature extraction networks and how they impact the object detection models accuracy.

*1) Libra R-CNN*: We have compared Libra R-CNN [18] with different backbones. This comparison reveals that the act of changing backbones with a solid feature aggregation model changes the performance. Regarding, Libra R-CNN with ResNeXt-101 as a backbone on top of the quality range. The two last models based on ResNet-50 and ResNet-101 as backbones, Libra R-CNN based ResNet-101 gain the highest performance (see Fig. 8).

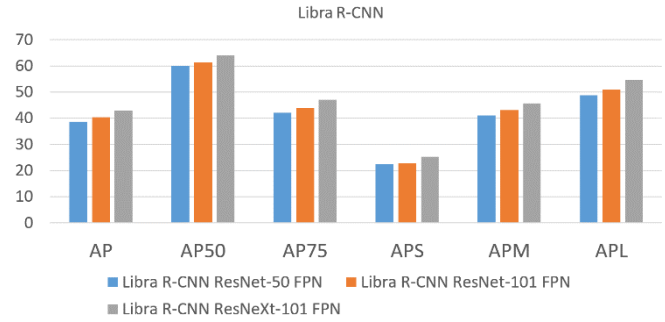


Fig. 8. Libra R-CNN Comparison based Different Feature Aggregation Models.

*2) Faster R-CNN:* Faster R-CNN [8] relying on ResNext-101-64x4d as a backbone and AugFPN as a feature aggregation model are leading the performance in this category. By fixing ResNet-50 as a backbone with changing different feature aggregation, the model based on AdaFPN gains the highest performance. Moreover, by fixing AugFPN and changing ResNext-101 the best performance was gained by ResNext-101-64x4d (see Fig. 9).

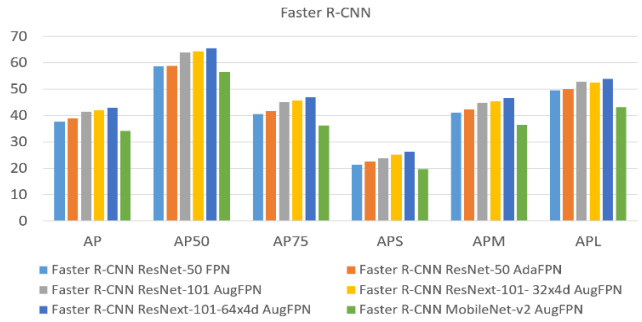


Fig. 9. Faster R-CNN Comparison based Different Feature Aggregation Models.

*3) FCOS:* The highest performance was obtained by FCOS [28] on the head, ResNext-101 as a backbone, and FPN as a feature aggregator model. By changing feature aggregation models FPN, AdaFPN, and AugFPN, moreover fixing ResNet-50 the AdaFPN gains the best performance in this category, after that FPN and finally AugFPN (see Fig. 10).

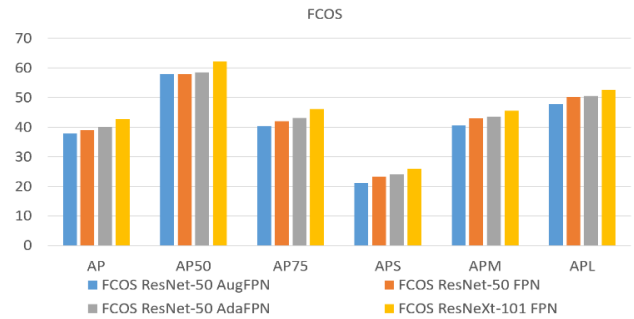


Fig. 10. FCOS Comparison based Different Feature Aggregation Models.

*4) Mask R-CNN:* Regarding Mask R-CNN [9] models based on a diversity of backbones and necks relied on our category, ResNet-101 and FPN combination leads the performance then, ResNeXt-101 and FPN. By fixing ResNet-101, mutating feature aggregation models the highest performance was gained by AugFPN, then FPN, and finally A2FPN. Concerning ResNet-50 as a backbone and A2 FPN or AugFPN as feature aggregation models, AugFPN attain the greatest performance (see Fig. 11).

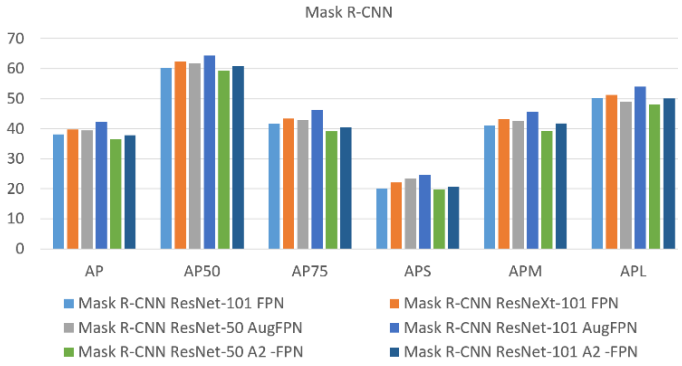


Fig. 11. Mask R-CNN Comparison based Different Feature Aggregation Models.

*5) HTC:* Related to HTC [33] model, ResNeXt-101 and A2FPN are leading in performance, the second performant fusion is ResNeXt-101 and FPN. Regarding the models based on ResNet as a backbone, ResNet-50 with A2FPN works better than ResNet-50 with FPN in terms of performance (see Fig. 12).

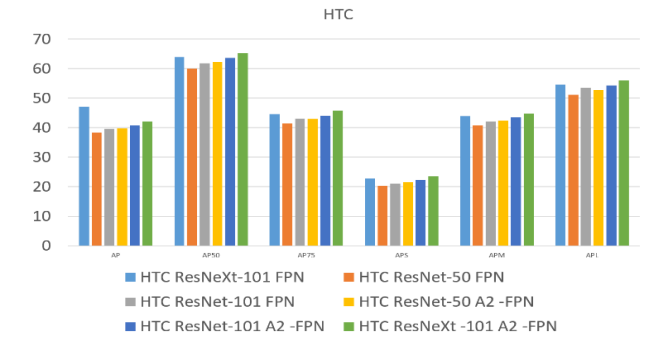


Fig. 12. HTC Comparison based Different Feature Aggregation Models.

*6) Cascade R-CNN:* Cascade R-CNN [29] performance was led by merging ResNet-101 and AC-FPN. The combination of ResNet-101 as a backbone and FPN neck has gained less performance (see Fig. 13).

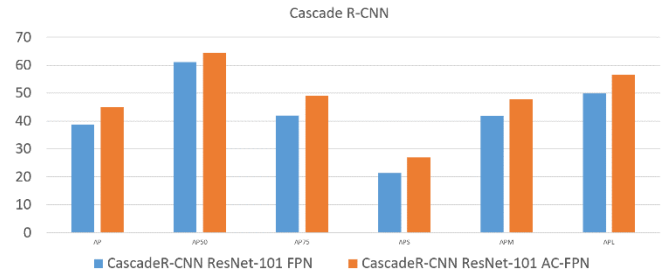


Fig. 13. Cascade R-CNN Comparison based Different Feature Aggregation Models.

*7) RetinaNet:* Regarding RetinaNet,[30] firstly, ResNeXt-101 as a backbone and FPN as a feature aggregation model compared to the other fusions, it has gained the highest performance; secondly, by merging ResNet-101 and FPN; and thirdly, ResNet-50 with AugFPN gains the performance, and finally, MobileNet-V2 with AugFPN (see Fig. 14).

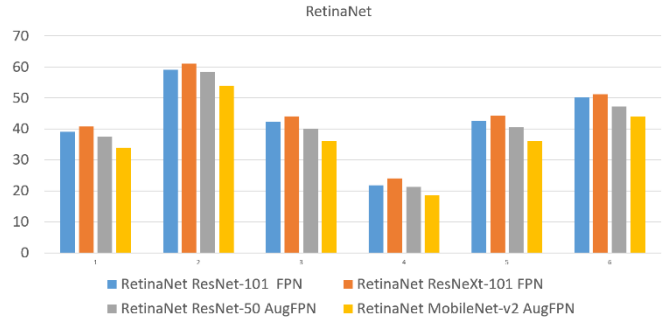


Fig. 14. RetinaNet Comparison based Different Feature Aggregation Models.

*8) Six Top average precision:* On the one hand, after extracting the 6 best models in terms of average precision, we have preferred to compare the methods that gain the top average precision. On the other hand, in terms of performance and based on our spider, centerNet2 achieves the best performance. The best method is based on Res2Net101-DCN as a backbone and BiFPN as a feature aggregation model. The second rank is for DetectRs based on ResNeXt-101-DCN as a backbone and RFP as feature extraction (see Fig. 15).

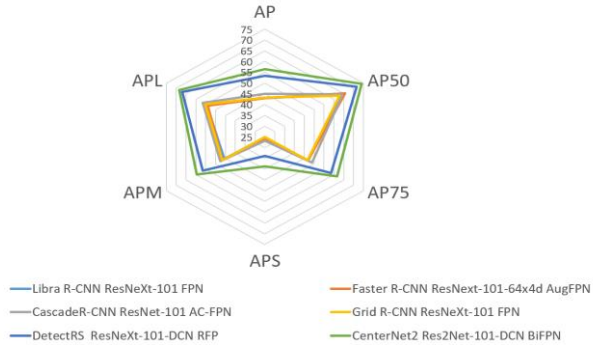


Fig. 15. Multicriteria Comparison based Different Feature Aggregation Models.

## VI. Discussion

In this paper, we have systematically depicted the importance of object detection components, covering the deep learning methodologies used in object detection, including, Two Stage detectors and one stage detectors.

Firstly, we have started by presenting object detection methodologies that have been categorized on traditional methods and based deep learning methodologies. Secondly, we have talked about the main arrangement of object detection based on deep learning that includes a backbone usually pretrained used to extract feature then feature aggregation model for merging high and low features called neck and finally, the head used for prediction.

Relied on our comparative study, we notice that the CenterNet2 with Res2Net-101-DCN as a backbone and BiFPN as a feature fusion model leads the performance and gains widespread dominance because of its supremacy regarding all criteria.

DetectRS with ResNeXt-101-DCNas a backbone and RFP as a feature fusion model is reaching the second score. HTC is gaining the third position with its high performance based on ResNeXt-101 as a backbone and FPN. We notice also that there is no intersection between all the compared algorithms, each algorithm gains its performance regarding all criteria that the underlying algorithm.

This comparison has also been made based on a set of criteria. The scores for each method evaluated were calculated using the Weight Score Model. Various scores or results have not only helped us determine an overall ranking, but they have also shown their internal strengths and weaknesses concerning each criterion.

This comparison has also revealed the importance of making a benchmark in order to have a global straightforward view of building efficient models with high performance.

One the one hand, we hold in mind that from this review and comparison study that object detection based deep learning models, backbone, neck and head, impacting highly the performance. On the other hand, generally, more used layers give high performance.

## VII. CONCLUSION

From the study handed, it has been noticed that several scientists and researchers from a diversity of ethnicities are working day after day on the object detection field, due to its utmost importance. Several models are appearing every month with the growth of deep learning.

This comparison could be used as a support, by handing researchers a scientific comparison of different object detection methodologies and their main models, in order to build performant models.

A comparison of neck used for feature aggregation between high and low features has been presented. We have been interested in giving you different necks and analyse the performance of their global models.

Future work will be focusing on the implementation of some of the different models of object detection-based deep learning. We aim to implement, test, and analyze the results.

## REFERENCES

[1] S. Bouraya and A. Belangour, "Object Detectors" Convolutional Neural Networks backbones : a review and a comparative study," vol. 9, no. 11, pp. 1379–1386, 2021.

[2] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 8759–8768, 2018, doi: 10.1109/CVPR.2018.00913.

[3] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-Janua, pp. 936–944, 2017, doi: 10.1109/CVPR.2017.106.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.

[5] W. Liu et al., "SSD: Single shot multibox detector," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9905 LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.

[6] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint triplets for object detection," Proc. IEEE Int. Conf. Comput. Vis., vol. 2019-Octob, pp. 6568–6577, 2019, doi: 10.1109/ICCV.2019.00667.

[7] R. Girshick, "Fast R-CNN," Proc. IEEE Int. Conf. Comput. Vis., vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.

[8] S. Ren, K. He, and R. Girshick, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," pp. 1–9.

[9] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

[10] S. Sharma, R. Kiros, and R. Salakhutdinov, "Action Recognition using Visual Attention," pp. 1–11, 2015, [Online]. Available: http://arxiv.org/abs/1511.04119.

[11] A. Kar, N. Rai, K. Sikka, and G. Sharma, "AdaScan: Adaptive scan pooling in deep convolutional neural networks for human action recognition in videos," Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017, vol. 2017-January, pp. 5699–5708, 2017, doi: 10.1109/CVPR.2017.604.

[12] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. M. Snoek, "VideoLSTM convolves, attends and flows for action recognition," Comput. Vis. Image Underst., vol. 166, pp. 41–50, 2018, doi: 10.1016/j.cviu.2017.10.011.

[13] N. Ballas, L. Yao, C. Pal, A. Courville, and R. Convolution, "D ELVING D EEPER INTO C ONVOLUTIONAL N ETWORKS," pp. 1–11, 2016.

[14] A. Karpathy and T. Leung, "Large-scale Video Classification with Convolutional Neural Networks."

[15] J. Donahue, "Long-term Recurrent Convolutional Networks for Visual Recognition and Description," 2014.

[16] N. Ballas, H. Larochelle, and A. Courville, "Describing Videos by Exploiting Temporal Structure," pp. 4507–4515, 2015.

[17] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," pp. 1–8.

[18] J. Pang, K. Chen, J. Shi, H. Feng, W. Ouyang, and D. Lin, "Libra R-CNN: Towards balanced learning for object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, no. 2, pp. 821–830, 2019, doi: 10.1109/CVPR.2019.00091.

[19] G. Ghiasi, T. Y. Lin, and Q. V. Le, "NAS-FPN: Learning scalable feature pyramid architecture for object detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 7029–7038, 2019, doi: 10.1109/CVPR.2019.00720.

[20] N. Wang et al., "NAS-FCOS: Fast Neural Architecture Search for Object Detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 11940–11948, 2020, doi: 10.1109/CVPR42600.2020.01196.

[21] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," CVPR, vol. 7, no. 3, pp. 1251–1258, 2014, doi: 10.4271/2014-01-0975.

[22] A. Vaswani et al., "Attention is all you need," Adv. Neural Inf. Process. Syst., vol. 2017-Decem, no. Nips, pp. 5999–6009, 2017.

[23] X. Wang and R. Girshick, "Non-local Neural Networks."

[24] Y. Chen, "A 2 -Nets : Double Attention Networks," no. NeurIPS, 2018.

[25] H. L. Fu Jun, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, "Dual Attention Network for Scene Segmentation."

[26] Y. Chen, M. Rohrbach, Z. Yan, S. Yan, J. Feng, and Y. Kalantidis, "Graph-Based Global Reasoning Networks," vol. 1.

[27] M. Tan, R. Pang, and Q. V Le, "EfficientDet : Scalable and Efficient Object Detection," pp. 10781–10790.

[28] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," Proc. IEEE Int. Conf. Comput. Vis., vol. 2019-Octob, pp. 9626–9635, 2019, doi: 10.1109/ICCV.2019.00972.

[29] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into High Quality Object Detection," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 6154–6162, 2018, doi: 10.1109/CVPR.2018.00644.

[30] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 318–327, 2020, doi: 10.1109/TPAMI.2018.2858826.

[31] C. F. Mykhailo and S. Alexander, "RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free."

[32] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, "Grid R-CNN," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 7355–7364, 2019, doi: 10.1109/CVPR.2019.00754.

[33] K. Chen et al., "Hybrid task cascade for instance segmentation," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2019-June, pp. 4969–4978, 2019, doi: 10.1109/CVPR.2019.00511.

[34] S. Qiao, L.-C. Chen, and A. Yuille, "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution," 2020, [Online]. Available: http://arxiv.org/abs/2006.02334.

[35] X. Zhou, V. Koltun, and P. Krähenbühl, "Probabilistic two-stage detection," 2021, [Online]. Available: http://arxiv.org/abs/2103.07461.