Convolutional Neural Networks Backbones for Object Detection

Ayoub Benali Amjoud(✉) and Mustapha Amrouch

IRF-SIC Laboratory, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
a05.benali@gmail.com, m.amrouch@uiz.ac.ma

Abstract. Detecting objects in images is an extremely important step in many image and video analysis applications. Object detection is considered as one of the main challenges in the field of computer vision, which focuses on identifying and locating objects of different classes in an image. In this paper, we aim to highlight the important role of deep learning and convolutional neural networks in particular in the object detection task. We analyze and focus on the various state-of-the-art convolutional neural networks serving as a backbone in object detection models. We test and evaluate them in the common datasets and benchmarks up-to-date. We Also outline the main features of each architecture. We demonstrate that the application of some convolutional neural network architectures has yielded very promising state-of-the-art results in image classification in the first place and then in the object detection task. The results have surpassed all the traditional methods, and in some cases, outperformed the human being's performance.

Keywords: Object detection · Convolutional neural networks · Review

1 Introduction

Detecting objects in a scene proved to be a very difficult task, which has been investigated for a variety of applications in recent years, such as face detection, self-driving cars, medical disease detection, video surveillance, and for natural disaster protection. The convolutional neural networks (CNNs) represent the heart of state-of-the-art object detection methods. They are used for extracting features. Several CNNs are available, for instance, AlexNet, VGGNet, and ResNet. These networks are mainly used for object classification task and have evaluated on some widely used benchmarks and datasets such as ImageNet (Fig. 1). In image classification or image recognition, the classifier classifies a single object in the image, outputs a single category per image, and gives the probability of matching a class. Whereas in object detection, the model must be able to recognize several objects in a single image and provides the coordinates that identify the location of the objects. This shows that the detection of objects can be more difficult than the classification of images.

Traditional object detection models tend to use methods such as Haar-Like features [1], HOG [2], and Scale-Invariant Feature Transform [3] for extracting the features in the image. Those approaches have been based on the way we could manually design the features or the model according to our understanding. Recently it has been proven

A. El Moataz et al. (Eds.): ICISP 2020, LNCS 12119, pp. 282–289, 2020.
https://doi.org/10.1007/978-3-030-51935-3_30

that it is more efficient to let the machine handle these tasks. And this is when the convolutional neural networks came to take control, achieving impressive successes [4, 5]. The present paper will be structured as follows. First, we review the leading state-of-the-art convolutional neural network architectures used in object detection. We then introduce the image datasets we have used to compare the networks along with the experiments. We further report the results of each architecture when used with state-of-the-art object detection models.

2 Convolutional Neural Network Backbones

The selection of CNN architectures to be covered in this article is not made randomly, but according to their popularity and performance in different state of the art object detection models.

2.1 AlexNet

Krizhevsky et al. [4] in 2012, developed a convolutional neural network composed of 8 layers, where 5 are convolutional and 3 are fully connected. The network is called AlexNet. In comparison to LeNet-5, AlexNet [6] has more layers and contains around 60 million parameters. Rectified Linear Units (ReLUs) are used for the first time as activations in AlexNet instead of sigmoid and tanh activations to add non-linearity. AlexNet is used in object detection models such as R-CNN [7], and HyperNet [8].

2.2 VGG-16

In 2014 a network called VGG-16 [9] was released, composed of 13 convolutional and 3 fully connected layers with ReLU activation. VGG-16 provides more layers compared to AlexNet and uses smaller filters of 2 × 2 and 3 × 3. It includes 138 million parameters. A deeper version of VGG called VGG-19 is available. VGG-16 is one of the most used architectures in object detection and achieved interesting performances; it's used for instance in algorithms like Fast R-CNN [10], Faster R-CNN [11], HyperNet [8], RON384 [12], SSD [13] and RefineDet [14].

2.3 GoogLeNet

Also called Inception V1, GoogLeNet [15] is a small network developed by Szegedy et al. in 2014. Their method is different from that of VGGNet and AlexNet. They came up with a new notion known as blocks of inception, where it embeds multi-scale convolutional transformations. The inception block includes filters of varying sizes 1 × 1, 3 × 3 and 5 × 5. It employs a 1 × 1 convolution in the middle of the network to reduce dimensionality and they opted to use global average pooling instead of fully connected layers. The network is made of 22 layers with 5 million parameters. GoogLeNet mainly is used in YOLO [16] object detection model.

2.4 ResNets

Convolutional neural networks have become more and more deeper with the addition of layers, but once the accuracy gets saturated, it quickly drops off. To solve this issue, He et al. in 2015 developed ResNets [5] which are based on residuals or skip connections. They also use Batch Normalization [17]. ResNets are mainly consisting of convolutional and identity blocks. There are many variants of ResNets, for instance, ResNet-34, ResNet-50 which is composed of 26 million parameters, ResNet-101 with 44 million parameters and ResNet-152 which is deeper with 152 layers. ResNet-50 and ResNet-101 are used widely in object detection models. While ResNet-50 is used in some object detection frameworks such as BlitzNet [18] and RetinaNet [19]. ResNet-101 is used in Faster R-CNN [5], R-FCN [20], and CoupleNet [21], etc.

2.5 Inception-ResNet-V2

Szegedy et al. published in 2016, Inception-ResNet-V2 [22], a CNN inspired by the ResNet and based on a hybrid approach by combining Inceptions and ResNet architectures, which use residual connections as an alternative to concatenation filters. Inception-ResNet-V2 is composed of 164 deep layers and about 55 million parameters. The Inception-ResNet models have led to better accuracy performance at shorter epochs. Inception-ResNet-V2 is used in Faster R-CNN G-RMI [23], and Faster R-CNN with TDM [24] object detection models.

2.6 DarkNet-19

A network developed to be small and efficient at the same time. It is based on many previous ideas like the Darknet reference, Network In Network [25], Inception [15, 26] and Batch Normalization [17]. Darknet-19 [27] uses convolutional layers instead of fully connected layers. It is composed of 19 convolutional and 5 max-pooling layers. It uses only 3×3 convolutional kernels and several 1×1 convolutional kernel to reduce the number of parameters. DarkNet-19 is used in YOLOv2 [27].

3 Data, Experiments and Results

3.1 Data

To assess the different CNNs mentioned above, we used several common data sets in the field of classification and object detection. First, we used the ImageNet database [28], one of the largest databases available today, it contains more than 14 million images from different categories. We used ImageNet to calculate Top-1 and Top-5 accuracy rates. Afterward, we used Pascal VOC [29] (2007 and 2012), and the Common Object in Context (COCO) [30] dataset for the object detection purposes.

Fig. 1. Examples of images from the ImageNet 2012 dataset.

3.2 Experiments and Results

In this section, we experiment with the CNNs mentioned in this paper along with the object detection models based on these networks under the different datasets and benchmarks. In Table 1, with the exception of the DarkNet-19, all the experiments are carried out with PyTorch[1], an open-source machine learning framework and Nvidia T4 GPU. The input size resolution is 224 × 224 for all networks except for Inception-ResNet-V2 where the input size is 299 × 299. To evaluate the computational complexity of each network we use the Multiply-And-Accumulate (MAC) operation that could be considered as two separate floating-point operations (FLOPs) [31]. In Table 2 and Table 3, the detectors are trained on Pascal VOC07 trainval and Pascal VOC12 trainval. The +S suffix means that the model is trained also for segmentation and extra annotations. For the Table 4, the models are trained on MS COCO trainval35k set.

Table 1. Network's performance on the ImageNet 1-crop accuracy rates.

Network	Params(M)	MACs(G)	Top-1 accuracy	Top-5 accuracy
AlexNet	61.1	**0.72**	56.55	79.09
VGG-16	138.36	15.5	71.59	90.38
GoogLeNet	6.62	1.52	69.78	89.53
ResNet-50	25.56	4.12	76.15	92.87
ResNet-101	44.55	7.85	77.37	93.56
Inception-ResNet-V2	55.84	13.22	**80.3**	**95.1**
DarkNet-19[a]	–	–	72.9	91.2

[a] https://pjreddie.com/darknet/imagenet/#darknet19

Table 2. Comparative results on Pascal VOC 2007 test set (%).

Detector	Backbone	Data	mAP
HyperNet [8]	AlexNet	07++12	65.9
PFPNet-R512 [32]	VGG-16	07++12	82.3
YOLOv1 [16]	GoogLeNet	07++12	63.4
BlitzNet512 [18]	ResNet-50	07++12+S	81.5
CoupleNet [21]	**ResNet-101**	07++12	**82.7**

[1] https://pytorch.org/.

Table 3. Comparative results on Pascal VOC 2012 test set (%).

Detector	Backbone	Data	mAP
PFPNet-R512 [32]	VGG-16	07++12	80.3
YOLOv1 [16]	GoogLeNet	07++12	57.9
CoupleNet [21]	**ResNet-101**	07++12	**80.4**

Table 4. MS COCO test-dev 2015 detection results (%).

Detector	Backbone	Data	mAP@.5	mAP@ [.5,.95]
PFPNet-R512 [32]	VGG-16	trainval35k	**57.6**	35.2
BlitzNet512 [18]	ResNet-50	trainval35k	50.9	32.5
CoupleNet [21]	**ResNet-101**	trainval35k	57.5	**36.4**
Faster R-CNN G-RMI [23]	Inception-ResNet-V2	trainval	55.5	34.7
YOLOv2 [27]	DarkNet-19	trainval35k	44.0	21.6

4 Discussion

In Table 1, we note that the Inception-ResNet-V2 network achieved a Top-1 Accuracy of 80.3% and 95.1% in the Top-5, higher than all other networks. Both ResNet-50 and ResNet-101 perform almost as well as Inception-ResNet-V2. Whereas AlexNet had 56.5% and 79.09% in Top-1 and Top-5 respectively, the remaining networks achieved nearly similar results. From Table 1, architectures based on the residual concept achieve better accuracy using a very reduced number of parameters compared to other architectures. For example, ResNet-50 has about 25 million parameters, ResNet-101 has around 44 million parameters and Inception-ResNet-v2 contains almost 55 million parameters, whereas VGG-16 has more than 138 million parameters. Although AlexNet and Inception-Resnet-V2 have a very similar number of parameters, the accuracy and number of MACs are much lower in AlexNet compared to Inception-Resnet-V2. Table 2 clearly shows that in object detection the networks with the best performance are VGG and ResNets. ResNet-101 with CoupleNet, ResNet-50 with BlitzNet512 and PFPNet-R512 with VGG-16 performed an accuracy of 82.7%, 81.5%, and 82.3% respectively in the Pascal VOC 2007 test set. In Pascal VOC 2012, Table 3 indicates that PFPNet-R512 with VGG-16 and CoupleNet with ResNet-101 achieved an accuracy of 80.3% and 80.4% respectively, while YOLOv1 with GoogLeNet achieved only an accuracy of 57.9%. For MS COCO we notice that the models based on VGG-16, ResNet-101 and Inception-ResNet-V2 achieved interesting results which are 57.6%, 57.5% and 55.5% respectively for mAP@.5, and 35.2%, 36.4% and 34.7% for the mAP@ [.5,.95]. While YOLOv2 with DarkNet-19 produced a mAP@.5 of 44% and a mAP@ [.5,.95] of 21.6%. According to the results obtained, we could mention the networks contributing to the highest performance in object detection are VGG-16, the ResNets family and also Inception-ResNets-v2, which combines the Inception and ResNets networks. This explains the wide use of these architectures in different object-detector models.

The following Table 5 shows the main features added for each architecture to improve the performance.

Table 5. Main added features in the CNNs.

Network	What's Novel?
AlexNet	- Apply Rectified Linear Units (ReLU) to add non-linearity
VGG-16	- Deep network, approximately twice as deep as AlexNet
GoogLeNet	- Using dense modules as opposed to stacking convolutional layers
ResNets	- Using batch normalization and skip connections
Inception-ResNet-V2	- Using residual inception blocks instead of inception modules - Adding the inception module (Inception-A) after the stem module - Using more inception modules
DarkNet-19	- Combine Darknet extraction, Network In Network, Inception and Batch Normalization in a single model

5 Conclusion

In this paper, we studied the state-of-the-art CNNs for object detection. We devoted the study to networks that have achieved remarkable performance. We outlined the datasets used for testing the CNNs as well as the object detection models. We compared those networks and models on multiple benchmarks and datasets. We report that the application of convolutional neural networks in object detection has given impressive state-of-the-art results.

Acknowledgements. This work is carried out with the support of the 'Centre National pour la Recherche Scientifique et Technique (CNRST)' of Morocco, through the Research Excellence Awards Program 2019.

References

1. Lienhart, R., Maydt, J.: An extended set of Haar-like features for rapid object detection. In: Proceedings of the International Conference on Image Processing, pp. I-900–I–903. IEEE, Rochester (2002). https://doi.org/10.1109/ICIP.2002.1038171
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), pp. 886–893. IEEE, San Diego (2005). https://doi.org/10.1109/CVPR.2005.177
3. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. Int. J. Comput. Vis. **60**, 91–110 (2004). https://doi.org/10.1023/B:VISI.0000029664.99615.94
4. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012)

5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE, Las Vegas (2016). https://doi.org/10.1109/CVPR.2016.90
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**, 84–90 (2017). https://doi.org/10.1145/3065386
7. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv:1311.2524 [cs]. (2013)
8. Kong, T., Yao, A., Chen, Y., Sun, F.: HyperNet: towards accurate region proposal generation and joint object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 845–853. IEEE, Las Vegas (2016). https://doi.org/10.1109/CVPR.2016.98
9. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs] (2014)
10. Girshick, R.: Fast R-CNN. arXiv:1504.08083 [cs] (2015)
11. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans. Pattern Anal. Mach. Intell. **39**, 1137–1149 (2017). https://doi.org/10.1109/TPAMI.2016.2577031
12. Kong, T., Sun, F., Yao, A., Liu, H., Lu, M., Chen, Y.: RON: reverse connection with objectness prior networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5244–5252. IEEE, Honolulu (2017). https://doi.org/10.1109/CVPR.2017.557
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. arXiv:1512.02325 [cs]. 9905, 21–37 (2016). https://doi.org/10.1007/978-3-319-46448-0_2
14. Zhang, S., Wen, L., Bian, X., Lei, Z., Li, S.Z.: Single-shot refinement neural network for object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4203–4212. IEEE, Salt Lake City (2018). https://doi.org/10.1109/CVPR.2018.00442
15. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9. IEEE, Boston (2015). https://doi.org/10.1109/CVPR.2015.7298594
16. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: unified, real-time object detection. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788. IEEE, Las Vegas (2016). https://doi.org/10.1109/CVPR.2016.91
17. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. arXiv:1502.03167 [cs] (2015)
18. Dvornik, N., Shmelkov, K., Mairal, J., Schmid, C.: BlitzNet: A Real-Time Deep Network for Scene Understanding. arXiv:1708.02813 [cs] (2017)
19. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. arXiv:1708.02002 [cs] (2018)
20. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Lee, D.D., Sugiyama, M., Luxburg, U.V., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems 29, pp. 379–387. Curran Associates, Inc. (2016)
21. Zhu, Y., Zhao, C., Wang, J., Zhao, X., Wu, Y., Lu, H.: CoupleNet: coupling global structure with local parts for object detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 4146–4154. IEEE, Venice (2017). https://doi.org/10.1109/ICCV.2017.444

22. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. arXiv:1602.07261 [cs] (2016)
23. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3296–3297. IEEE, Honolulu (2017). https://doi.org/10.1109/CVPR.2017.351
24. Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A.: Beyond Skip Connections: Top-Down Modulation for Object Detection. arXiv:1612.06851 [cs] (2016)
25. Lin, M., Chen, Q., Yan, S.: Network In Network. arXiv:1312.4400 [cs] (2014)
26. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision. arXiv:1512.00567 [cs] (2015)
27. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster, Stronger. arXiv:1612.08242 [cs] (2016)
28. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575 [cs] (2015)
29. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes challenge: a retrospective. Int J Comput Vis. **111**, 98–136 (2015). https://doi.org/10.1007/s11263-014-0733-5
30. Lin, T.-Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context. arXiv:1405.0312 [cs] (2015)
31. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated Residual Transformations for Deep Neural Networks. arXiv:1611.05431 [cs] (2017)
32. Kim, S.-W., Kook, H.-K., Sun, J.-Y., Kang, M.-C., Ko, S.-J.: Parallel feature pyramid network for object detection. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018, pp. 239–256. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01228-1_15