2022 12th Iranian/Second International Conference on Machine Vision and Image Processing (MVIP)
Department of Electrical and Computer Engineering, Shahid Chamran University of Ahvaz, Iran, 23 & 24 February 2022

2022 12th Iranian/Second International Conference on Machine Vision and Image Processing (MVIP) | 978-1-6654-1216-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/MVIP53647.2022.9738779

A face detection method via ensemble of four versions of YOLOs

Sanaz Khalili
Department of Computer Sciences
Vali-e-Asr University of Rafsanjan
Rafsanjan, Iran
khalili.sanaz94@gmail.com

Ali Shakiba
Department of Computer Sciences
Vali-e-Asr University of Rafsanjan
Rafsanjan, Iran
ali.shakiba@vru.ac.ir

***Abstract*—We implemented a real-time ensemble model for face detection by combining the results of YOLO v1 to v4. We used the WIDER FACE benchmark for training YOLOv1 to v4 in the Darknet framework. Then, we ensemble their results by two methods, namely, WBF (Weighted boxes fusion) and NMW (Non-maximum weighted). The experimental analysis showed that the mAP increases in the WBF ensemble of the models for all the easy, medium, and hard images in the datasets by 7.81%, 22.91%, and 12.96%, respectively. These numbers are 6.25%, 20.83%, and 11.11% for the NMW ensemble.**

Keywords— YOLO, Face detection, WBF, NMW, Ensemble

I. INTRODUCTION

Traditional computer vision tasks include object detection, image processing, image recognition, and image classification [1]. The object detection problem is central in computer vision. A solution for this problem is then used to develop solutions for other problems, e.g. image segmentation [2]. In this paper, we implement a real-time model made of an ensemble of the YOLO v1 through v4[3]–[6] to solve the face detection problem. The problem of face detection is the first step in various more sophisticated problems involving face identification, face recognition, and facial expression analysis [7].

There are two different approaches for object detection in deep learning: (1) the regression such as SSD [10] and YOLO [11], and (2) the R-CNN series such as Faster R-CNN [9] and Fast R-CNN [8]. The YOLO is an effective algorithm for object detection problems [12]. This algorithm only looks once at an image and then, detects all of the objects in that image. The YOLO also predicts bounding boxes and their classes in real-time, and all is done at one stage. Various generalizations of the YOLO algorithm are proposed such as RCNN and DPM [13].

A. Our contribution

In this paper, we trained the YOLOv1 through the YOLOv4 for the WIDER FACE dataset using the Darknet framework. The Darknet is a fast open-source framework written in C and CUDA for training deep neural networks with GPU and CPU [14]. These four trained models are ensembled with non-maximum weighted (NMW) [15] and weighted-boxes-fusion (WBF) [16] approaches. For WBF, our ensemble model increase the mAP by 7.81%, 22.91%, and 12.96% for the easy, medium, and hard instances of the WIDER dataset, respectively, compared with the best mAP for each YOLOv1 to v4 alone. These numbers are 6.25%, 20.83%, and 11.11% for NMW ensemble.

II. RELATED WORK

A. YOLOv1

The YOLOv1 is the first version of the YOLO proposed in 2016 by Joseph Redmon in [3] by borrowing the idea of the GoogleNet [17] architecture. The YOLOv1 architecture consists of twenty-four convolution layers followed by four pooling layers. Also, as the last network layer in YOLOv1, two fully connected layers are employed, where a Dropout was present between the two fully connected layers. In all layers except the final layer, the activation function is the leaky ReLU. In the last layer, the sigmoid function is used as the activation function. The YOLOv1 receives an image of size 448×448×3 as input and produces a grid of size 7×7×30 as output. In YOLOv1, the mean squared error of all predictions is used as the loss function. Each prediction includes the coordinates of the bounding boxes, the class probabilities, and a confidence score.

B. YOLOv2

The YOLOv2 is proposed by Joseph Redmon in 2016 [4] and used the DarkNet19 at its core. The YOLOv2 network architecture includes 23 convolutional layers followed by five layers of max-pooling. The activation function for all layers is leaky ReLU. The Batch normalization technique is used in training this network. This model improves the YOLOv1's performance by incorporating new ideas and predicting more bounding boxes than the YOLOv1. It also used the anchor boxes and fine-grained features in the network. The input of the YOLOv2 is an image of size 416×416×3, and its output is a 13×13×40 grid.

978-1-6654-1216-2/22/$31.00 ©2022 IEEE

C. YOLOv3

The YOLOv3 [5] uses the Darknet with 53 layers and incorporates skip connections, residual connections, downsampling, and upsampling. The Darknet53 does the feature extraction task in the YOLOv3. In YOLOv3, another 53 layers of the convolutional network are added after the Darknet53. The input to the YOLOv3 is a 416×416×3 image, and its output is a grid of sizes 13×13×255, 26×26×255, and 52×52×255.

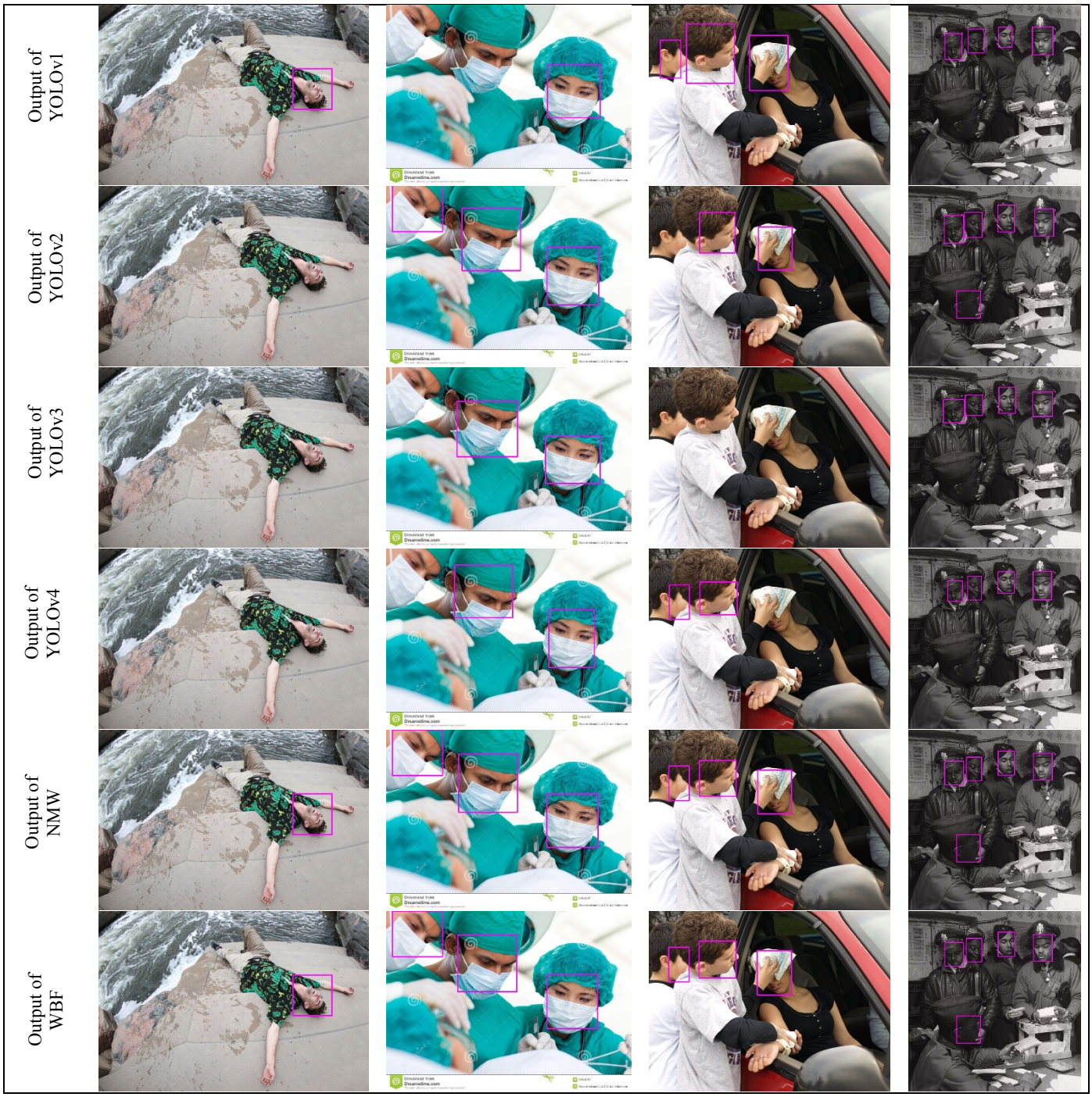

Figure 1. The output of some random instances of the WIDER FACE dataset for each YOLOv1 to v4 as well as the output of their NMF and WBF ensembles. As it can be visually verified by these samples, there are cases where a face is detected by only one YOLO model. So, their ensemble is expected to have a better performance (which is the case as it is given in Table 4).

D. YOLOv4

THE YOLOV4 MODEL USES CIOU LOSS, DROPBLOCK REGULARIZATION, CMBN, MOSAIC DATA AUGMENTATION, MISH ACTIVATION, SAT, CSP, AND WRC [6]. THESE CHANGES MAKE THE YOLOV4 BETTER AND MORE ACCURATE THAN THE PREVIOUS VERSIONS.

E. Ensemble

In an ensemble, the outputs of multiple models are combined. Usually, the generalization of the models in this technique is significantly improved [18]. We use two methods to obtain the ensemble of the YOLOs v1 to v4: (1) non-maximum weighted (NMW), and (2) weighted-boxes-fusion (WBF). In both methods, we need to determine the weights of their combination for every single YOLO model. This weight characterizes the effectiveness of the models in the combination.

III. PROPOSED METHOD

In the proposed method, we first train each of the four YOLO models independently. Then, the output bounding boxes in every four models are combined by WBF and NMW methods. More details on the training of each network are given in Table 1. After training, the averaged boxes in the ensembled model are constructed by utilizing the confidence score for each predicted bounding box in the YOLOv1 to v4 models in the WBF, as is suggested in [16]. We did the same for the NMW as well as the elimination of the repetitive bounding boxes according to [15].

Table 1. Technical details for training models

Algorithm name	batch	Image size	iteration
YOLOv1	64	448×448	9500
YOLOv2	64	416×416	6500
YOLOv3	64	416×416	7000
YOLOv4	32	416×416	2500
YOLOv4	16	608×608	2700

The base idea behind choosing these four models and ensembling their output is as follows: (1) increasing the number of bounding boxes which the ensembled model can predict. (2) taking the advantages of one model where other models fail to detect faces. For example, there is a face on the first column in Figure 1 that only YOLO v1 could detect, whereas the other YOLO versions could not. The maximum number of bounding boxes that each model can predict is given in Table 2.

Table 2. The maximum number of bounding boxes in each model

Algorithm name	Maximum number of bounding boxes in each model
YOLOv1	7×7×2=98
YOLOv2	13×13×5=845
YOLOv3	52×52×3+26×26×3+13×13×3=10647
YOLOv4	19×19×3+38×38×3+76×76×3=22743
NMW, WBF	98+845+10647+22743=34333

IV. EXPERIMENTAL EVALUATIONS

In this section, we compare the NMW and WBF ensembles of the YOLOs v1 to v4 using the mAP measure for the WIDER FACE dataset. We did the training of these models using the Darknet framework [14] on the Google Collaboratory infrastructure using an Nvidia K80 GPU.

The WIDER FACE dataset is an standard and big dataset for face detection problem with 32,203 images. There are many challenges in this dataset due to scale, occlusion, and pose variety. The WIDER FACE dataset is divided into 3 parts, namely the easy, the medium, and the hard images [19].

The number of iterations for training the YOLOv1 was more than the other versions of YOLOs. The YOLOv2, v3, and v4 use the anchor box compared to YOLOv1, which causes faster convergence rates. We used the mean average precision (mAP) to evaluate the performance of every single method for the WIDER FACE dataset. The mAP is one of the frequently used metrics for comparing the performance of object detection models. These mAP values are given in Table 3.

According to Table 3, the YOLOv3 has the biggest mAP and hence has the best performance. This is due to its better capability in the prediction of the bounding boxes. Note that we trained the YOLOv4 twice: (1) the first time, we trained it for input images of size 416×416 with batch size 32, and (2) then, we increased the size of the input, however with a smaller batch size. This smaller size of the batch was due to the memory limitation of the Google Collaboratory free service. By the way, this increase in the size of the input image for YOLOv4 caused an increase in the mAP. This happened due to the fact that the model views a bigger image. The second configuration of the YOLOv4 was especially useful for the hard set of test images in the WIDER FACE dataset, where there are many small bounding boxes.

Table 3. The calculated mAP for each YOLOs v1-4 and the NMW and WBF ensembles.

Algorithm name	mAP Easy	mAP Medium	mAP Hard
YOLOv1	0.18	0.11	0.06
YOLOv2	0.42	0.26	0.20
YOLOv3	0.58	0.41	0.43
YOLOv4-416	0.63	0.44	0.48
YOLOv4-608	0.64	0.48	0.54
NMW	0.68	0.58	0.60
WBF	**0.69**	**0.59**	**0.61**

The YOLOv1 has the smallest output among all these four YOLOs, it has just one output of size 7x7. It also has the smallest depth among these four versions of YOLOs. In some sense, this might be useful in object detection, since the model is required to detect objects with the smallest number of features. On the other hand, the increased mAP metric for the YOLOv2 in comparison with the YOLOv1 is due to its bigger output size as well as employing the anchor boxes. The same applies to the YOLOv3 since it not only uses the anchor boxes and has a bigger output size, but has more depth and extracts the features in three stages. The YOLOv4 even has a greater mAP because the activation function is changed to mish as well as the size of the output grid is bigger. However, as it can be visually verified in Figure 1, there are cases that an older version of these four YOLOs can predict a bounding box whereas others fail to do that. Hence, we expect their ensemble has greater mAP, which happens to be true.

In both cases of the NMW and WBF, the mAP increased. We tested many different combinations of weights and different skip thresholds. Finally, we choose the IoU threshold of 0.6, a skip threshold of 0.1 with weights 0.1, 0.5, 0.8, 0.9 for YOLO v1 to v4, respectively. With this setting, we were able to reach the biggest mAP values, which are given in Table 4. Note that with this setting, the weight for YOLOv4 is bigger than the weights for the other YOLOs. This is because YOLOv4 has the biggest mAP among all four YOLO versions. On the other hand, the YOLOv1, which has the smallest mAP, has the smallest weight in the ensemble. In comparison, the mAP for the WBF ensemble technique is bigger than the NMW in this setting.

For WBF, our ensemble model increase the mAP by 7.81%, 22.91%, and 12.96% for the easy, medium ,and hard instances of the WIDER dataset, respectively, compared with the best mAP for each YOLOv1 to v4 alone. These numbers are 6.25%, 20.83%, and 11.11% for the NMW ensemble.

V. Conclusion

In this paper, we combined the outputs of four versions of the YOLO models by WBF and NMW methods for the WIDER FACE benchmark. We evaluated these two approaches and compared their performance using the well-known mAP metric. Both ensemble methods provided more accuracy in terms of mAP, compared to the output of every individual YOLO model. Between the two approaches of the ensemble, the WBF has the biggest mAP than the NMW, which means that the WBF performs better. For WBF, our ensemble model increase the mAP by 7.81%, 22.91%, and 12.96% for the easy, medium, and hard instances of the WIDER dataset, respectively, compared with the best mAP for each YOLOv1 to v4 alone. These numbers are 6.25%, 20.83%, and 11.11% for the NMW ensemble.

As future work, we tend to ensemble YOLOv5 with the rest of the models using the test-time-augmentation (TTA) techniques, similar to the approach in [20].

References

[1] X. Jiang, T. Gao, Z. Zhu, and Y. Zhao, "Real-time face mask detection method based on yolov3," *Electron.*, vol. 10, no. 7, pp. 1–17, 2021, doi: 10.3390/electronics10070837.

[2] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–22, 2021, doi: 10.1109/TPAMI.2021.3059968.

[3] 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.

[4] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-Janua, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.

[5] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018, [Online]. Available: http://arxiv.org/abs/1804.02767.

[6] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," Apr. 2020, [Online]. Available: http://arxiv.org/abs/2004.10934.

[7] C. Li, R. Wang, J. Li, and L. Fei, *Face detection based on YOLOv3*, vol. 1031 AISC. Springer Singapore, 2020.

[8] 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.

[9] Y. Wang and J. Zheng, "Real-time face detection based on YOLO," *1st IEEE Int. Conf. Knowl. Innov. Invent. ICKII 2018*, vol. 2, pp. 221–224, 2018, doi: 10.1109/ICKII.2018.8569109.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN : Towards Real-Time Object Detection with Region Proposal Networks," pp. 1–14.

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

[12] P. D. Jędrzej Świeżewski, "What is YOLO Object Detection?," 2020, [Online]. Available: https://appsilon.com/object-detection-yolo-algorithm/.

[13] R. Girshick, F. Iandola, T. Darrell, and J. Malik, "Deformable part models are convolutional neural networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 437–446, 2015, doi: 10.1109/CVPR.2015.7298641.

[14] J. Redmon, "Darknet: Open source neural networks in C," 2018. https://pjreddie.com/darknet/.

[15] H. Zhou, Z. Li, C. Ning, and J. Tang, "CAD: Scale Invariant Framework for Real-Time Object Detection," *Proc. - 2017 IEEE Int. Conf. Comput. Vis. Work. ICCVW 2017*, vol. 2018-Janua, pp. 760–768, 2017, doi: 10.1109/ICCVW.2017.95.

[16] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," Oct. 2019, doi: 10.1016/j.imavis.2021.104117.

[17] C. Szegedy *et al.*, "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.

[18] T. Zhou, H. Lu, Z. Yang, S. Qiu, B. Huo, and Y. Dong, "The ensemble deep learning model for novel COVID-19 on CT images," *Appl. Soft Comput.*, vol. 98, p. 106885, 2021.

[19] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-Decem, pp. 5525–5533, 2016, doi: 10.1109/CVPR.2016.596.

[20] C. Gonzalo-Martín, A. García-Pedrero, and M. Lillo-Saavedra, "Improving deep learning sorghum head detection through test time augmentation," *Comput. Electron. Agric.*, vol. 186, p. 106179, 2021, doi: 10.1016/j.compag.2021.106179.