An Efficient Deep Learning Framework for Face Mask Detection in Complex Scenes

Sultan Daud Khan, Rafi Ullah, Mussadiq Abdul Rahim, Muhammad Rashid, Zulfiqar Ali, Mohib Ullah, Habib Ullah

An efficient deep learning framework for face mask detection in complex scenes

Sultan Daud Khan[1], Rafi Ullah[1], Mussadiq Abdul Rahim[1], Muhammad Rashid[1], Zulfiqar Ali[1], Mohib Ullah[2], and Habib Ullah[3]

[1] Department of Computer Science, National University of Technology, Islamabad, Pakistan.
[2] Norwegian University of Science and Technology, 2815 Gjøvik, Norway.
[3] Faculty of Science and Technology, Norwegian University of Life Sciences, 1430 Ås, Norway.

Abstract. COVID-19 has caused a global health crisis that has infected millions of people across the globe. Currently, the fourth wave of COVID-19 is about to be declared as Omicron. The new variant of COVID-19 has caused an unprecedented increase in cases. According to World Health Organization, safety measures must be adopted in public places to prevent the spread of the virus. One effective safety measure is to wear face masks in crowded places. To create a safe environment, government agencies adopt strict rules to ensure adherence to safety measures. However, it is difficult to manually analyze the crowded scenes and identify people violating the safety measures. This paper proposed an automated approach based on a deep learning framework that automatically analyses the complex scenes and identifies people with face masks or without face-masks. The proposed framework consists of two sequential parts. In the first part, we generate scale aware proposal to cover scale variations, and in the second part, the framework classifies each proposal. We evaluate the performance of the proposed framework on a challenging benchmark data set. We demonstrate that the proposed framework achieves high performance and outperforms other reference methods by a considerable margin from experimental results.

Keywords: Face mask detection · deep learning · multi-scale object proposals · fully convolutional neural network

1 Introduction

COVID-19 has been spreading exponentially across the globe, causing more than 5 million deaths now. Due to the lack of proper medical treatment, World Health Organization (WHO) has recommended preventive measures, including wearing masks and maintaining social distance to control the further spread of the disease. For this purpose, government agencies have adopted strict policies to ensure the adoption of preventive measures in public places. However, it is a complicated and tedious job to manually monitor a large number of people in an unconstrained environment. Therefore, in this work, we propose an artificial

intelligence-based system that automatically detects and counts the number of people without wearing masks. We believe that the proposed system will support government agencies in predicting the outbreak of COVID-19 by using statistical data obtained by the proposed system.

Face-mask detection is a computer vision problem involving finding faces with masks on in images and videos. Face-mask in images can be easily identified with the human eye. However, the problem of detecting face-mask is challenging for computers due to the complex dynamics of human faces. Due to significant variation in scales [1], poses [2], and appearances of human heads [3], it remains a challenging problem to detect face-mask in complex scenes with high precision and recall rates. Therefore, the goal of the face-masks detector is to detect faces with different orientations, poses, illumination, skin colour, and appearances. Convolutional neural networks (CNN) based deep learning models can easily address the problem of variations in pose and appearance in complex images since these models are inherently transnational invariant. However, CNN based deep learning models can not handle the variations in scales and sizes of objects in natural images, which becomes a challenging problem for the object detectors. Since variation in object scales naturally occurs in natural images, it is essential to address this problem precisely to detect objects of different scales. In this work, we employ a deep learning network and feed the network with a lot of training data to learn the best representative features of face masks. The input to our face-mask detector is an image, and the output is the bounding boxes around the head with a confidence score.

Most traditional methods treat the head detection problem as a particular case of object detection. Generally, the basic architecture of these models follows the pipeline of two-stage detectors, i.e., generation of object proposal followed by classification and regression stage. Object proposal generation network is the pre-processing step and mainly affects the performance of object detectors. The sliding window approach has been adopted by most object detectors that generate object proposals of different sizes and scales for each pixel. This strategy leads to computational cost since it will generate many proposals. This strategy also leads to the accumulation of many bounding boxes around a single object, which lowers the object detector's precision and recall rates. To address these problems, several methods have been reported in the literature to generate the optimum number of object proposals while increasing the precision and recall rates. For example, EdgeBox [4] exploits edge information to generate object proposals that precisely hypothesize an object's location. DeepBox [5] proposed a method that re-ranks the object proposals using a bottom-up strategy to improve the objectness. Deep Proposal [6] builds an inverse cascade network that uses the initial and final layers of the convolutional neural network to hypothesize the promising location of the object. Multi-Box [7] hypothesize the object location by employing bounding box regression.

Usually natural videos and images are complex, where similar objects are different from each other in terms of scales and sizes. For example, the size of objects near the camera appears large, while the size of the same object appears

small at a distance. Due to these problems, current two-stage object detectors face challenges detecting an object of various scales. We present a novel strategy for generating object proposals that will capture a wide range of variations in object scales to address this problem. Precisely, we propose a scale-aware object detection framework that follows the traditional pipeline of object detection and consists of the following sequential modules:

1. The first module is the object proposal generation network, which generates multi-scale object proposals to capture the scale distribution of the heads that appeared in an image. Briefly, we train a head/background binary classifier using a fully convolutional network (FCN) [8] on image patches with annotated heads. This network can take an image of arbitrary size and output a dense heat map or objectness map. Each pixel of the heat map shows the probability of the presence of a head and background. For this purpose, we exploit FCN to generate object proposals of different scales and sizes by first generating an image pyramid of multiple levels. Each level of the pyramid corresponds to a copy of the original image of a different resolution. We then feed each pyramid level to FCN and obtain a heat map. The size of the heat map is equal to the size of the corresponding level. After generating heat maps of multiple resolutions, we then resize the feature maps to the same size and employ the non-maximal suppression method to suppress low confidence locations and generate object proposals for the classification stage
2. The second module is the detection module that takes each proposal as an input and provides each proposal to two sibling branches. The first branch classifies each proposal into pre-defined classes, and the second branch is the regression branch that predicts the coordinates of bounding boxes.

We summarize the contribution of proposed framework as follows:

- We exploit a Fully Convolutional Network (FCN) to generate high-quality multi-scale object proposals.
- The proposed object proposal generation strategy reduces the computational cost and increases precision and recall rate compared to traditional approaches.
- We evaluate the performance of the proposed framework on challenging datasets, and from experiment results, we demonstrate the effectiveness of the proposed framework
- We compare the proposed framework with statistical and deep learning models and demonstrate that the proposed framework outperforms other reference methods.

2 Proposed Methodology

In this section, we provide the details of the proposed methodology. As shown in Figure 1, the input to the framework is an image of arbitrary size, and the output is the set of bounding boxes over the people wearing face-mask. Generally,

the proposed framework consists of two sequential networks. The first network is the object proposal generation network. The object proposal generation network is responsible for generating multi-scale proposals of various sizes. The second network is the classification network that classifies each proposal (generated during the first stage) into two classes, i.e., a person with a face mask or without a face mask. We generate multi-scale object proposals by first generating an image

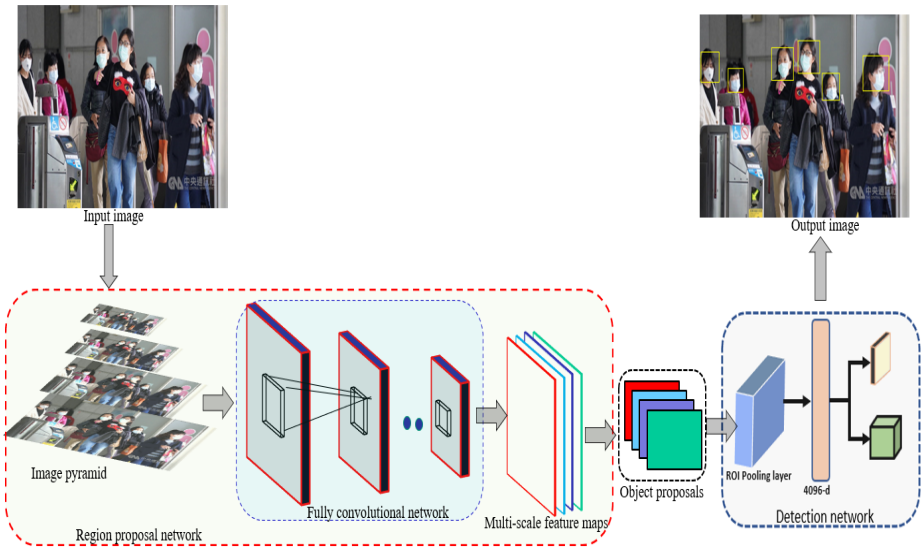

Fig. 1. Pipeline of proposed face-mask detector framework

pyramid of N number of levels and then providing each level of the pyramid to the object proposal network as an input. The object proposal network generates a heat map that corresponds to each level. We then apply a non-maximum suppression algorithm to suppress the background and obtain the regions belonging to people wearing face masks.

2.1 Object proposal network

The proposed object proposal generation network is based on VGG16 [9], which is a widely adopted model for image classification tasks. The proposed object proposal generation network consists of a stack of five convolutional blocks. The first convolutional block $Block_1$ consists of two convolutional layers, with a filter size of 3 x 3 and stride 1. The reason behind using the smallest filter size is to capture more delicate details and preserve the spatial resolution of the feature maps. Similarly, the second convolutional block $Block_2$ consists of two convolutional layers with a filter size of 3 x 3 and a stride of 1. While $Block_3, Block_4, Block_5$ consist of three convolutional layers. The feature map from each convolutional

block is passed through the max-pooling layer of filter size 2 x 2 and stride 2. Max-pooling layer reduces the resolution of the input feature map and reduces the parameter to increase the training process. The feature maps are then passed through a stack of three fully convolutional layers of size 1 x 1. We obtain a feature map for each level of the pyramid and then apply the non-maximum suppression method to obtain high confidence object proposals of different sizes. We then apply the ROI-pooling layer to normalize object proposals that will re-size object proposals to make them fit for the detection network.

Difference with other networks: Our proposed object proposal generation network is similar to Region Proposal Network (RPN) [10]. However, it differs in the following ways. RPN uses the feature map of the last convolutional layer for generating object proposals. Due to the large receptive field, the last convolutional layer can cover large objects, while the information about the small objects is lost. This is due to the reason that RPN is not able to detect small objects in complex scenes. Furthermore, RPN uses a limited and fixed object scale set $\{128, 256, 512\}$ with limited aspect ratios of $\{1:1, 1:2, 2:1\}$ for detecting multi-scale objects in an image. Such limited scale sets are inadequate for detecting multi-scale objects in complex scenes. Compared to large objects, the size and scale of the human head vary significantly in crowded scenes. However, (RPN) [10] due to limited scale range, can not cover all ranges of scales. Our proposed object proposal network is different from RPN because the proposed network is independent of anchors and generates object proposals specific to the object class.

2.2 Training

For training the object proposal network, we use a patch-wise training scheme. For this purpose, we divide the image into K number of overlapped patches. We then compute Intersection-over-Union (IoU) for each patch. IoU measures the percent of overlap between the given patch and ground truth and computed as $\text{IoU} = \frac{\Omega_p \cap \Omega_g}{\Omega_p \cup \Omega_g}$, where Ω_p is the patch of an image and Ω_g is the ground truth. We select patches as positive samples for which $\text{IoU} \geq 0.5$ and the rest of the patches are treated as negative samples. Since the background primarily occupies natural images, the number of negative patches will be much larger than positive patches. This creates a data imbalance problem that affects the generalization capability of the network, as the network will be getting more biased towards the negative samples/background. To increase the number of positive samples, we crop several patches around the human head and adopt a data augmentation technique [11] to generate different variants of positive patches.

We use Xavier technique [12] for initializing the weights of the network and use the stochastic gradient descent (SGD) strategy with a learning rate of 0.01 for training the network. We gradually decrease the learning process by using the strategy adopted in [13]. We train the network for 100 epochs on Titan-V GPU with 12GB RAM and use a batch size of 512 samples.

2.3 Detection network

After obtaining object proposals using the network discussed in section 2.1, we then employ a detection network that classifies each proposal into pre-defined classes and predicts the bounding boxes. Our detection network re-size multi-scale object proposals by employing an ROI pooling layer and then feeding each object proposal to two different branches. The first is the classification branch, which classifies the object proposals into two categories. The second branch is the regression branch that predicts the bounding boxes of the input proposals.

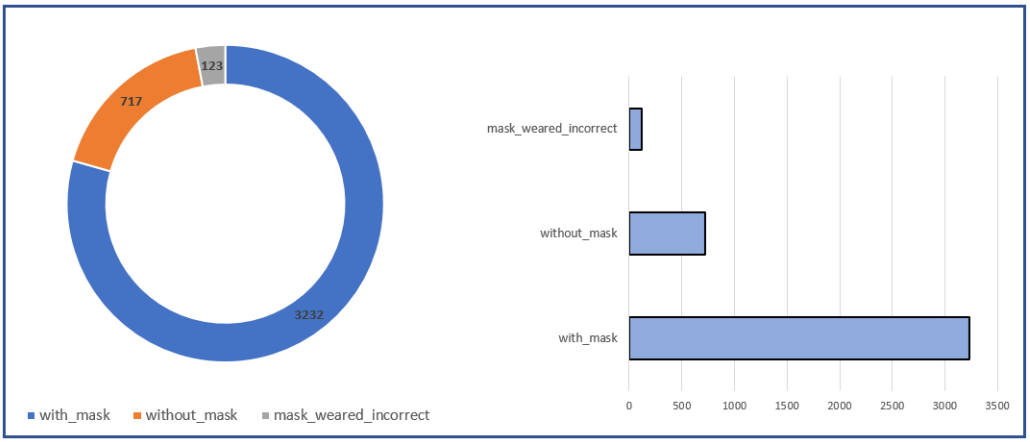

Fig. 2. Distribution of dataset among three classes.

3 Experiment Results

This section provides the details of the dataset used for evaluating and comparing the proposed framework with other related methods. We also discuss the various metrics used for the evaluation.

Mask Dataset: This dataset is collected by MakeML community and is publicly available on the link: https://makeml.app/datasets/mask. The dataset consists of different images captured from different scenes. The images are of different resolutions, and the human head has a wide range of variations in scale, size, orientation, illumination, and appearance. The dataset consists of 4072 images and three classes, i.e., with_mask, without_mask, and mask_weared_incorrect. The distribution of images among these three classes is shown in Figure 4. From Figure 4, it is obvious that the dataset contains a large number of images (3232) belonging to the with_mask class, 717 images belong to without_mask and only 123 images to mask_weared_incorrect class. We divide the dataset into two splits,

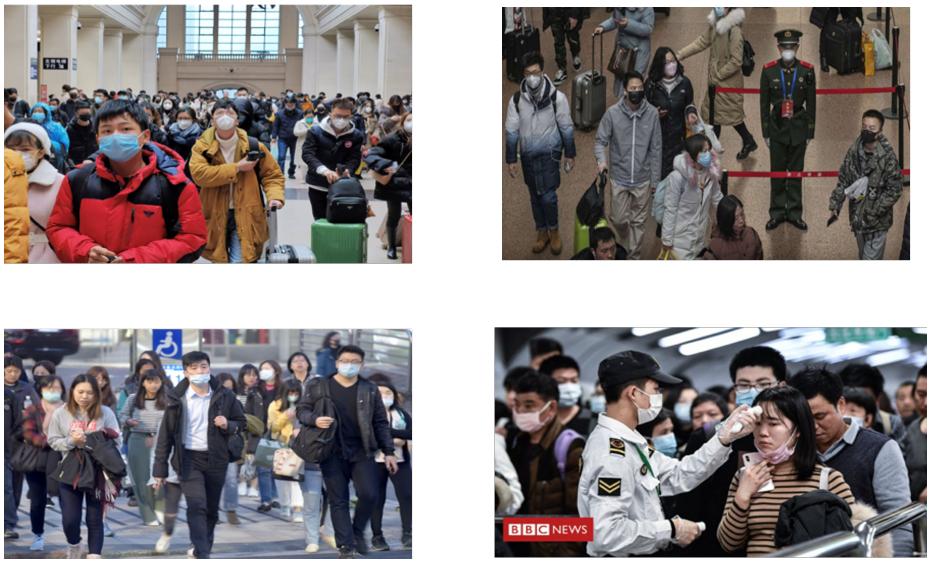

Fig. 3. Illustrates some random samples of the dataset

i.e., training and testing. We randomly select 70% images for training and the rest of 30% for testing. Few samples from the dataset are shown in Figure 3.

To quantitatively evaluate and compare the performance of the proposed framework with other related methods, we use the mean average precision (mAP) metric, which is widely used for evaluating the performance of object detectors. Mean average precision (mAP) computes a score and measures how well the detector predicts the object's location by comparing the predicted bounding box and ground truth. Mean average precision is calculated as follows:

$$mAP = \frac{1}{N} \sum_{c=1}^{k=N} AP_c \tag{1}$$

Where N is the total number of classes, and AP_c is the average precision of class c. We now evaluate and compare the proposed framework with other reference methods. For comparisons, we divide the experiment set up into two phases. In the first phase, we compare the proposed framework with hand-crafted feature models, while in the second phase, we perform a comparison with deep learning models. During the first phase of the experiment, we use three different versions of the Viola-Jones algorithm [14] based on the feature selection, namely, VJ-LBP, VJ-HOG, VJ-CRF [15]. We implement these models in MATLAB using built-in functions. In addition to these variants, we use DPM [16] and FFD [17] for comparisons. For DPM, we use the code available on https://github.com/rbgirshick/voc-dpm and for FFD, we use code available on https://github.com/apennisi/fast_face_detector. We use these codes to train the model on

the mask dataset, and the obtained results are then compared with the proposed framework. The results of these models are reported in Table 1.

Table 1. Comparison of proposed method with hand-crafted feature models

Methods	**mAP @ 0.5**	**mAP @ 0.7**
VJ-LBP	0.39	0.25
VJ-HOG	0.37	0.21
VJ-CRF	0.41	0.32
DPM-Head	0.53	0.47
FFD	0.46	0.39
Proposed	0.75	0.68

From Table 1, it is evident that the proposed framework beats other reference methods by a significant margin. We observe from experiment results that variants of Viola-Jones face difficulty in detecting heads and faces with different orientations. We observed that these models are very sensitive to noise and illumination, and as a result, these models produce many false positives that further degrade their performance. FFD also utilizes the learning pipeline of Viola-Jones and achieves good performance compared to other variants of Viola-Jones. This is because FFD re-ranks the score of the bounding boxes and adopts a multi-view detection strategy to accumulate channel features. Furthermore, we observe that DPM-Head can detect medium or large heads. However, it faces difficulty detecting small heads of size ($\leq$25 x 25 pixels).

We choose Faster R-CNN, R-CNN, SSD, YOLO, and Mask-RCN to compare the proposed framework's performance with other deep learning models. We fine-tunned these models on the mask dataset. To comprehensively analyze the performance of these models on the mask dataset, we use different backbone networks, for example, VGG16, AlexNet, and ZF. Furthermore, we also evaluate the performance of the proposed framework with these backbone networks.

We report the results of these models in Table 2. From Table 2, it is evident that the proposed framework achieves good results compared to other deep learning models. Yolo and SSD achieve relatively low performance. Since these models use the feature map of the last convolutional layer, these models face challenges in detecting small heads that negatively affect the performance. On the other hand, the proposed framework captures significant variation in object scales by generating scale-aware proposals and accurately detecting heads in complex scenes.

4 Conclusion

We proposed an efficient detection model to detect people with face masks and without face masks. The proposed framework utilizes a fully convolutional network to generate scale-aware proposals to cover scale variations naturally occurring in images. We perform experiments on challenging benchmark datasets. We

Fig. 4. Visualization of results generated by the proposed method. Detected bounding boxes are labeled with the confidence score. (best view is zoomed in

Table 2. Comparison of proposed method with deep learning models

Models	**Deep models**	**mAP @ 0.5**
Faster R-CNN [10]	ZF [18]	0.71
	VGG16 [9]	0.72
	AlexNet[19]	0.67
R-CNN [20]	ZF [18]	0.66
	VGG16 [9]	0.68
	AlexNet[19]	0.63
YOLO [21]	13-layered architecture	0.62
SSD [7]	VGG16 [9]	0.58
Proposed	ZF [18]	0.73
	VGG16 [9]	0.75
	AlexNet[19]	0.69

evaluate the performance of the proposed framework in both quantitative and qualitative ways. The experiment results demonstrate that the proposed framework beats other reference methods by a significant margin. In future work, we further enhance the performance of the proposed framework and deploy the framework in real-time.

References

1. Sultan Daud Khan, Ahmed B Altamimi, Mohib Ullah, Habib Ullah, and Faouzi Alaya Cheikh. Tcm: Temporal consistency model for head detection in complex videos. *Journal of Sensors*, 2020, 2020.
2. Mohib Ullah, Habib Ullah, and Ibrahim M Alseadonn. Human action recognition in videos using stable features. 2017.
3. Mohib Ullah, Mohammed Ahmed Kedir, and Faouzi Alaya Cheikh. Hand-crafted vs deep features: A quantitative study of pedestrian appearance model. In *2018 Colour and Visual Computing Symposium (CVCS)*, pages 1–6. IEEE, 2018.
4. C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.
5. Weicheng Kuo, Bharath Hariharan, and Jitendra Malik. Deepbox: Learning objectness with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2479–2487, 2015.
6. Amir Ghodrati, Ali Diba, Marco Pedersoli, Tinne Tuytelaars, and Luc Van Gool. Deepproposal: Hunting objects by cascading deep convolutional layers. In *Proceedings of the IEEE international conference on computer vision*, pages 2578–2586, 2015.
7. Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
8. Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.
9. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

10. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.
11. Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.
12. Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.
13. Timur Garipov, Pavel Izmailov, Dmitrii Podoprikhin, Dmitry Vetrov, and Andrew Gordon Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8803–8812, 2018.
14. Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, 2001.
15. Xiaofeng Ren. Finding people in archive films through tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
16. Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.
17. Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li. Aggregate channel features for multi-view face detection. In *IEEE international joint conference on biometrics*, pages 1–8. IEEE, 2014.
18. Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
19. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
20. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
21. Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.