Impact of Multi-size Small Anchors and Penalized IoU on Face Mask Detection

Jieyu Chen[1, *, †]
[1]School of Mathematics
The University of Edinburgh
Edinburgh, United Kingdom
[*]J.Chen-216@sms.ed.ac.uk

Yexuan Gao[2, *, †]
[2]Faculty of Mathematics
University of Waterloo
Waterloo, Canada
[*]y348gao@edu.uwaterloo.ca

Xinyu Zhang[3, *, †]
[3]Computer Science Department
The Chinese University of Hong Kong
Hong Kong, China
[*]xyzhang0980@outlook.com

Minzhu Zhao [4, *, †]
[4]School of Communication
Hong Kong Baptist University
Hong Kong, China
[*]18251846@life.hkbu.edu.hk
[†]These authors contributed equally.

***Abstract*—Wearing masks as one of the most effective ways to diminish the transmission of COVID-19 increases the demand for automatic face mask detection in all countries. Face masks belong to the small objects category in images, thus introducing the challenge of training a robust face mask detector, particularly for small object detection. Feature Pyramids derived from deep convolutional neural networks are commonly used to achieve scale-invariant object detection; however, it does not reach the same level of performance in detecting face masks as in detecting larger objects. This work proposed two methods: fully utilizing the feature map extract from the neural network by adding small multiscale anchors on the last feature map, which contains the highest resolution information. The other is to replace the standard IoU calculation with a tolerant strategy for small objects. Using these two methods, we improve the accuracy of small object detection while increasing the general average precision.**

Keywords-Small object Detection; Feature Pyramid Network; Faster R-CNN; IoU

I. INTRODUCTION

As the COVID-19 pandemic began to wreak havoc worldwide from early 2020, wearing face masks, highly recommended by World Health Organization (WHO), is considered one of the most effective measures against COVID-19. During the global public health crisis, many countries count monitoring and evaluating masks wearing as a community mitigation strategy [1]. In this sense, training object detectors on the face mask dataset benefit the current electronic door lock and help mitigate public health emergencies.

Face masks belong to the "small objects" category with less than usual physical size in the real world; thus, they are difficult to be detected in digital images and videos. "Small objects" also refer to objects in images occupying an area less than or equal to 32^2 pixels [2]. However, dominant object detectors such as Faster R-CNN, Mask R-CNN, SSD, and YOLO do not perform well on small objects compared to corresponding larger objects [3].

Feature pyramids drive detecting objects on different scales. Although traditional feature pyramidal hierarchy or image pyramid results in high computational complexity and memory consumption, the inherent multiscale pyramidal hierarchy developed from convolutional neural networks reduces the cost and advanced scale-invariant object detection. However, detecting small objects is still a challenge in computer vision [3]. Intuitively, small objects augmentation is introduced to improve performance on detection [4]. But deep convolutional neural networks still find it hard to extract information from small objects. Aside from data augmentation, image inputs are more robust through generative adversarial networks (GAN) before performing object detection [5, 6]. Despite that, this approach introduces the challenge of training GANs. Contextual information is also used to provide more details in recognition [7, 8]. Still, it is only applicable in segmentations, making this approach impractically for some datasets abstract from real applications.

The traditional training approach of the Regional Proposal Network (RPN) makes the bottleneck in the performance of the Faster R-CNN model on small objects detection. Binary class labels are assigned to each anchor proposed by a sliding window faction. And the anchor is scored based on the Intersection-over-Union (IoU), defined as the intersection over the union of the areas of anchors and ground truth boxes. An anchor is assigned with a positive label given that its IoU is

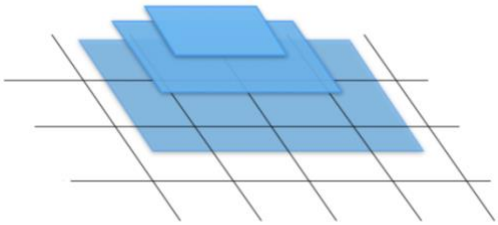

Figure 1. Multiscale small anchor boxes strategy. To increase small anchor boxes by adding small anchors in the P2 feature map.

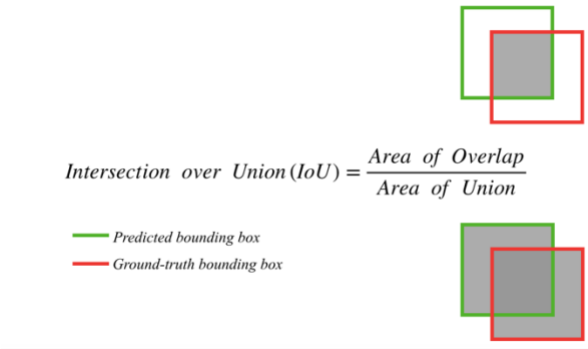

Figure 2. The intersection over union is obtained by dividing the area of overlap between the two bounding boxes by the union area.

Greater than 0.7, while giving a negative label with its IoU less than 0.3. However, the IoU is sensitive to the anchor size, and small objects are more difficult to capture by proposals rightly. On the contrary, a large object can tolerant more spatial offset and rank higher IoU.

To solve the inherent inadequate information of small objects and scale-dependent issue on RPN-based networks, we proposed two methods based on Faster R-CNN enhanced by FPN. The first idea is to generate smaller anchors on the bottom layer of the feature pyramid network (FPN), thus introducing more small anchors to feed into Fast R-CNN and RPN. Meanwhile, we proposed a tolerant strategy for small anchors in IoU calculation by assigning a factor dependent on the size of ground truth boxes.

We evaluated our methods on the Face Mask dataset [9], where small objects are in the majority. The face mask dataset results suggest that our two approaches enhance the precision on the small object while keeping the performance on medium and large objects, thus improving the total accuracy of object detection.

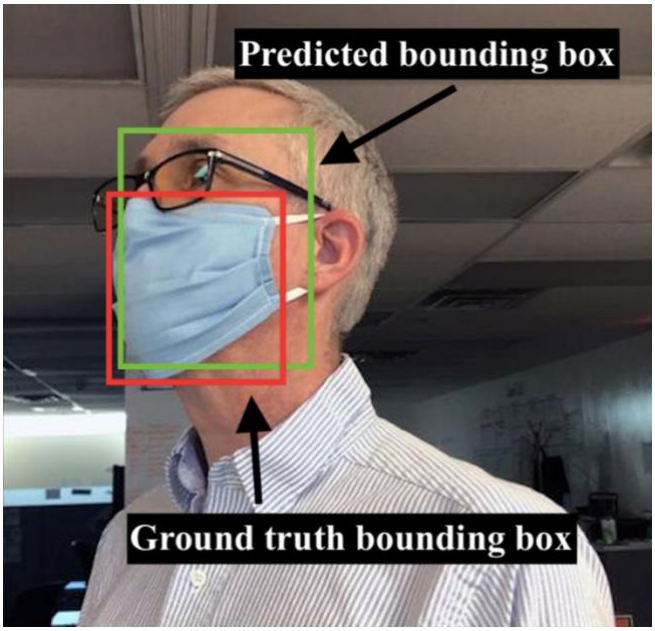

Figure 3. An example of detecting a mask in an image. The ground truth bounding box, a manually labeled bounding box to indicate the position of the objects, is drawn in red. The predicted bounding box, also known as the anchor box, is drawn in green. Shiraz Ahmed, "Homemade Face Masks: Washing, Wearing and Making Them", WDET, April 17, 2020, https://wdet.org/posts/2020/04/17/89502-homemade-face-masks-washing-wearing-and-making-them/

II. Proposed Model

A. *Adding Small Anchor Boxes in RPN*

Due to the small size of most target objects in our dataset, we propose adding more small anchor boxes in the Region Proposal Network of our model to capture small targets. In the Faster-RCNN + FPN network, the feature pyramid network for RPN contains a set of five feature maps (P2/ P3/ P4/ P5/ P6), and at each sliding-window location of the feature maps, several anchors are anticipated related with different scales and aspect ratios to capture the objects [10]. In our works, to add more and denser anchor boxes, we propose to add more small anchor boxes in P2, the feature map with the highest resolution. Thus, for each pixel in the P2, there are more anchor boxes created, which can increase the possibility of small object capturing.

In Fig. 1, we graphically present the proposed strategy adding small anchor boxes in the P2 feature map. The graph shows how the strategy increases the number of small anchors in FPN during training, leading to better performance in detecting small objects.

B. *Penalized Intersection over Union (PIoU)*

Intersection over Union (IoU) is a popular evaluation [11] metric used to measure the accuracy of an object detector. In Faster R-CNN, the IoU is attained by dividing the area of overlap between the ground truth bounding box and predicted

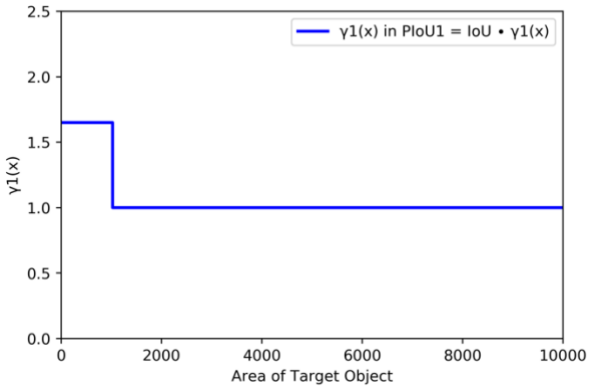

Figure 4. The sketch of $\gamma_1(x)$ in $PIoU1 = IoU \cdot \gamma_1(x)$.

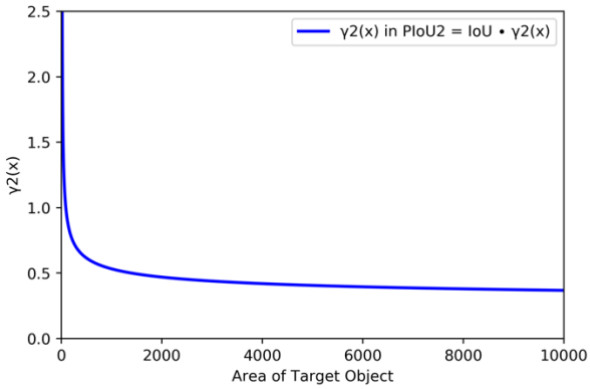

Figure 5. The sketch of $\gamma_2(x)$ in $PIoU2 = IoU \cdot \gamma_2(x)$.

bounding box by the area of the union of the two boxes, as shown in Fig. 2. And in Fig. 3, we can see the ground truth bounding box and predicted bounding box on detecting a mask in an image.

During training in Faster R-CNN, a predicted bounding box with high IoU (IoU > 0.7 or it has the largest IoU on the corresponding ground truth box) will be labeled 'positive' and passed to further anchor box regression to correct its position by the regression loss function [10]. However, this labeling procedure favors large objects. Large objects spanning multiple sliding-window locations often have a high IoU with many anchor boxes. In contrast, a small object may only be matched with a single anchor box with a low IoU [4]. As a result, the model is biased in favor of large and medium objects. The result of our baseline model can suggest that the mean Average Precision (mAP) of small objects is 39.7%, 42.4% lower than large objects.

To tackle this bias, we introduce the Penalized Intersection over Union (PIoU), which emphasizes detecting small objects:

$$PIoU = IoU \cdot \gamma(x), \tag{1}$$

where $\gamma(x)$ is a function dependent on ground truth box area x.

To determine the specific PIoU on our dataset, we propose to find appropriate constants (the special case of $\gamma(x)$) and functions to increase the percentage of small objects' anchor boxes to be labeled 'positive.' Then, more small anchor boxes can participate in the following training procedure, the model's bias towards large objects is also weakened.

$PIoU1$, shown in Fig. 4 is computed by:

$$PIoU1 = IoU \cdot \gamma_1(x), \tag{2}$$

where

$$\gamma_1(x) = min\left(\left\lfloor\frac{1024}{x}\right\rfloor, 1\right) \cdot 0.65 + 1. \tag{3}$$

$PIoU1$ is a new metric with the following properties:

1) For small objects ($x<32^2$), $PIoU1 = IoU \cdot 1.65$

2) For medium and large objects ($x>32^2$), $PIoU1 = IoU \cdot 1$

$PIoU2$, shown in Fig. 5 is computed by:

$$PIoU2 = IoU \cdot \gamma_2(x), \tag{4}$$

where

$$\gamma_2(x) = \frac{1.1922155}{log(x) - log\left(e^{7/4}\right)}. \tag{5}$$

$PIoU2$ is a new metric with the following properties:

1) $\gamma_2(x)$ converges to zero when the area of the ground truth box tends to be infinitely large.

2) For small objects, $\gamma_2(x)$ decreases rapidly as the size of objects increases, and it reaches 0.53 when the object is in size of 32^2.

3) For medium and large objects $\gamma_2(x)$ decreases slowly from 0.53 and gradually converges to 0.24 as objects get larger to the size of 800^2. When the size of objects is in the range of 50^2 to 125^2, $\gamma_2(x)$ is round to 0.4, then $PIoU2 = IoU \cdot 0.4$.

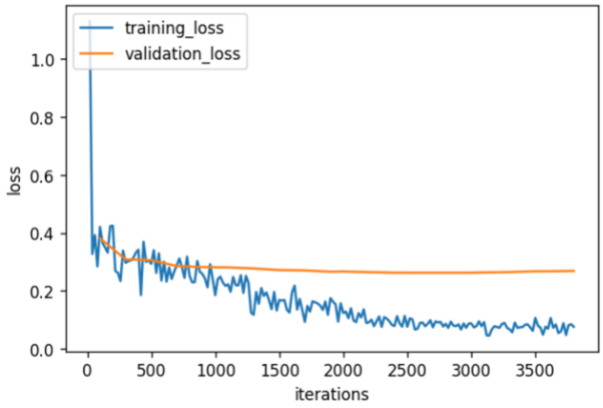

Figure 6. Loss of training set vs. loss of validation set.

C. *Faster R-CNN with FPN*

Current convolutional network architectures such as Faster R-CNN [10] and Faster R-CNN with FPN [12] have been utilized for object detection tasks. FPN is proposed to improve the task of multiscale object detection. The original architecture leverages the idea of utilizing semantic-strong feature maps analogous to feature image pyramids to give better results on extracting features than what is done in a regular RPN architecture without significantly increasing the cost of computational resources. However, this network architecture, which has improved performance in general prediction tasks, does not necessarily improve the performance of predicting small objects from an image from previous research [13]. Small objects are only recognizable in high-resolution images or feature maps. Their information is not represented well in higher-level feature maps due to the up-sampling behavior of the Feature Pyramid Network. Combining high-semantic feature maps with lower ones does not help with detecting small objects. We propose two methods based on the Faster RCNN with FPN architecture to improve performance on detecting small objects. We decrease the smallest anchor size in the lowest level of the feature maps to allow the model to have a sliding window search with smaller steps when proposing RoIs. We also investigate the method of giving preference to proposed anchors, particularly for smaller objects, in adjusting the calculation of IoU for objects with an area less than 32^2. Both methods are based on the idea of allowing RoIs, which possibly contain small objects, to be proposed more easily from feature maps. By increasing proposed areas that only contain relatively small objects, our methods can improve the performance of detecting small objects while preserving the behavior of the current model to medium and large objects.

III. Experiment Setup

A. *Baseline Setup*

We use Faster R-CNN with FPN implementation from Detectron2 [14] with a ResNet-50 backbone for the experiments. To explore our methods of detecting small objects, we choose a dataset in which most objects are categorized as small (< 32^2) in COCO format. Adapted from the original FPN architecture [12], we choose the anchor sizes for each feature map level to be [16,32,64,128,256] with the same scaling factor 2, rather than including the largest anchor size of 512^2 in the original design, due to that the images in this dataset all have the size of the longest edge less than 400px. We set the base learning rate to 0.05 with a warm-up scheduler and a step decay scheduler. Comparing the loss on the training set and the validation set, we select maximum iteration at 3800 according to the loss on the validation set and training set in Fig. 6 to maximize the model's performance on our dataset. Experiments are run on a single-thread NVIDIA Tesla T4 GPU.

B. *Adding Small Anchor Boxes in RPN*

In the first set of experiments, we investigate the effect of adding small anchor boxes in the RPN. We vary the anchor box sizes setting in feature pyramid maps. Six settings are tested firstly. We vary the setting of small-size anchor boxes added in P2 to find the setting with the best performance. In the first setting, a set of 8^2 and 16^2 anchor boxes are added in the P2. In the second setting, a set of 4^2,8^2 and 16^2 anchor boxes are added in the P2. In the third setting, a set of 2^2,4^2,8^2 and 16^2 anchor boxes are added in the P2. In the fourth setting, a set of 2^2,4^2,8^2 anchor boxes are added in the P2. In the fifth setting, a set of 2^2 and 4^2anchor boxes are added in the P2. In the final setup, only 2^2 anchor boxes are added in P2.

Furthermore, we add a set of small anchor boxes in different feature pyramid maps (P2/ P3/ P4/ P5/ P6) to investigate the effect of adding small anchor boxes in the different feature maps. Five settings are tested in the experiments. In the first setting, a set of 2^2,4^2,8^2 anchor boxes are added in P2. In the second setting, a set of 2^2,4^2,8^2 anchor boxes are added in P3. In the third setting, a set of 2^2,4^2,8^2 anchor boxes are added in P4. In the third setting, a set of 2^2,4^2,8^2 anchor boxes are added in P5. In the fourth setting, a set of 2^2,4^2,8^2 anchor boxes are added in P6. In the final setup, a set of 2^2,4^2,8^2 anchor boxes are added in all maps, including P2, P3, P4, P5, and P6.

C. *IoU adjustment*

The first set of experiments is designed for the special case of $PIoU = IoU \cdot \gamma(x)$, where $\gamma(x)$ is a constant. We investigate the effect of changing the value of $\gamma(x)$, the coefficient in $PIoU$. In order to find the coefficient with the best performance on small-object detection, 19 settings are tested in the experiments. In the first 5 settings, we try $\gamma(x) = 2, 3, 4, 5, 6$. Then in the following 14 experiments, we start taking $\gamma(x)$ from 1.45 and test every 0.05 until it reaches 2.10.

In the second set of experiments, $\gamma(x)$ is a subfunction of $PIoU = IoU \cdot \gamma(x)$. We construct and test 8 functions to explore the effect of different $\gamma(x)$ on our model's AP. In the first setting, a continuous function is applied to objects of all sizes. In the second and third setting, functions are designed for small objects specifically, while medium and large objects take $\gamma(x) = 1$ in $PIoU$. In the fourth and fifth settings, $PIoU = IoU \cdot 2.0$ and 1.65 for small objects, and $PIoU = IoU \cdot 1$ for others.

TABLE I. ANCHOR SIZES EXPERIMENTS IN P2

Settings of P2	Average Precision			
	Small	***Medium***	***Large***	***All***
Baseline[a]	39.729	65.876	82.137	51.873
[8,16]	40.094	65.223	84.105	52.717
[4,8,16]	40.403	65.407	83.474	52.431
[2,4,8,16]	**43.936**	66.314	87.316	**55.117**
[2,4,8]	39.976	65.178	88.980	52.407
[2,4]	40.638	65.787	82.623	52.360
[2]	38.655	64.385	79.916	50.285

a. Baseline has anchor size setting: [[16], [32], [64], [128], [256]]. Change anchor sizes in P2 for each experiment and keep P3~P6 the same.

TABLE II. SMALL ANCHOR BOXES EXPERIMENTS IN FPN

Settings	Average Precision			
	Small	***Medium***	***Large***	***All***
Baseline[a]	39.729	65.876	82.137	51.873
P2 [2,4,8,16]	**43.936**	66.314	87.316	**55.117**
P3 [2,4,8,32]	42.049	65.386	84.752	53.426
P4 [2,4,8,64]	39.830	65.986	83.921	51.747
P5 [2,4,8,128]	39.582	65.138	84.535	51.803
P6 [2,4,8,256]	38.461	65.671	79.518	50.679
P2~6 [2,4,8,16] [2,4,8,32] [2,4,8,64] [2,4,8,128] [2,4,8,256]	38.870	64.067	83.778	50.008

a. Baseline has anchor size setting: [[16], [32], [64], [128], [256]]. Adding anchor boxes 2^2, 4^2, 8^2 in P2~P6 respectively.

D. Anchor sizes and IOU adjustment

This experiment is to explore the impact on model performance when both previous methods are used. We selected the setup from the previous experiments, which yielded the best AP. The anchor sizes of P2 in FPN have been increased from 16^2 to 2^2,4^2,8^2,16^2, with the change of $PIoU = IoU$ multiplied by corresponding coefficients for small, medium, and large objects.

TABLE III. PIOU-CONSTANT EXPERIMENTS

Constant $\gamma(x)$	Average Precision			
	Small	***Medium***	***Large***	***All***
Baseline[a]	39.729	65.876	82.137	51.873
1.55	40.162	64.275	83.561	52.159
1.65	41.436	68.504	79.887	53.971
1.70	39.961	65.886	84.792	52.554
2.00	**42.023**	65.658	84.573	**53.795**

a. $PIoU = IoU \cdot \gamma(x)$, baseline has $\gamma(x) = 1.00$.

IV. RESULTS AND ANALYSIS

A. Adding Small Anchor Boxes in RPN

In Table I, we present the results using different settings for small anchor boxes in P2. The baseline is using a size of 16^2 anchor boxes in P2. When we added a set of 8^2 and 16^2 anchor boxes in P2 (the second row), there was no significant improvement in performance. When we added a set of 4^2,8^2,16^2 anchor boxes in P2 (the third row), there was no significant improvement in performance. When we added a set of 2^2,4^2,8^2,16^2 anchor boxes in P2 (the fourth row), the performance improved notably. When we added a set of 2^2, 4^2, 8^2anchor boxes in P2 (the fifth row), there was no significant improvement in performance. When we added a set of 2^2 and 4^2 anchor boxes in P2 (the sixth row), the performance improved slightly. When we only added a set of 2^2 anchor boxes in P2 (the seventh row), the performance degraded slightly. It shows that adding a set of 2^2,4^2,8^2,16^2 small anchor boxes in P2 is a more effective strategy.

The result reveals that adding multiscale small anchors on the last layer of the feature map with the highest resolution information generates more accurate proposals for small objects.

Table II presents the results of adding small anchor boxes in different feature pyramid maps in FPN. The first setting is adding a set of 2^2,4^2,8^2 anchor boxes in P2 (the first row). When we added a set of 2^2,4^2,8^2 anchor boxes in P3 (the second row), the performance degraded slightly. When we added a set of 2^2,4^2,8^2 anchor boxes in P4 (the third row), the performance degraded notably. When we added a set of 2^2,4^2,8^2 anchor boxes in P5 (the fourth row), the performance degraded notably. When we added a set of 2^2,4^2,8^2 anchor boxes in P6 (the fifth row), the performance degraded notably. When we added a set of 2^2,4^2,8^2 anchor boxes in all pyramid feature maps, including P2, P3, P4, P5, and P6 (the sixth row), the performance degraded notably. It shows that adding small anchors in P2 is a more effective strategy.

Even the bottom layer contains strong semantic values through the top-down pathway and lateral connection of feature maps from residual blocks. In this sense, using a feature map with more resolution information from FPN gives us more accurate areas to capture small objects.

TABLE IV. PIOU-FUNCTION EXPERIMENTS

Function $\gamma(x)$[a]	Average Precision			
	Small	*Medium*	*Large*	*All*
Baseline	39.729	65.876	82.137	51.873
$F_1(x)$[b]	**40.748**	62.012	79.824	50.573
$F_2(x)$[c]	37.535	62.578	83.054	49.427
$F_3(x)$[d]	39.510	63.288	84.738	51.055
$F_4(x)$[e]	39.613	63.137	84.608	50.591
$F_5(x)$[f]	**40.794**	62.152	85.335	51.447

a. $\gamma(x)$ in function represents the area of ground truth box.
b. $F_1(x) = \frac{1.1922155}{\log(x) - \log(e^{7/4})}$ for x>9,
c. $F_2(x) = 2 \times 1.1^{2^{\log_4(x)-3}-4}$ for x<32^2,
d. $F_3(x) = 2 \times 1.1^{2^{5-\log_4(x)}-4}$ for x<32^2,
e. $F_4(x) = \min\left(\left\lfloor\frac{1024}{x}\right\rfloor, 1\right) + 1$,
f. $F_5(x) = \min\left(\left\lfloor\frac{1024}{x}\right\rfloor, 1\right) \cdot 0.65 + 1$.

B. IoU adjustment

By finding an appropriate constant $\gamma(x)$ in PIoU (see table III), AP on small-object detection is enhanced. The best result is observed when $\gamma(x)$ takes 2.00 for all small objects, which AP for small objects is increased by 2.4%, overall AP is improved by 1.9% as well.

Table III shows that a tolerant strategy for small objects can improve the accuracy while preserving the precision for detecting objects of medium and large size.

The results of PIoU-function experiments are presented in Table IV. As we can see, the subfunction $F_5(x)$ performances the best with an increase of 1.1% in AP-Small, namely $\gamma_1(x)$ in $PIoU1 = IoU \cdot \gamma_1(x)$. In addition, the subfunction $F_1(x)$ also improves AP-Small by 1%, named it $\gamma_2(x)$. We introduced these two functions earlier in section II.B with their sketches and properties.

Applying a continuous tolerant function as a factor in IoU computation does not show an advantage over directly adding a constant, as shown in Table III; this might be because our dataset is relatively small, and boosting the IoU for small objects benefits only a set of small anchors.

C. Anchor sizes and IoU adjustments results

When combining previous adjustments on anchor sizes and IoU calculation, the model has higher AP and AP-Small than our baseline. Table V presents that AP-Small increases in applying anchor sizes change and IoU adjustment for small objects. The improvement in the second experiment, where we adjust both anchor sizes and IoU calculations, is equivalent to only adjusting anchor sizes. It has the highest APs and overall AP among all the model settings. From the experiments, we argue that the IoU calculation of anchor boxes and ground truth does not have an as significant impact as adding small anchors in FPN in improving the model ability of small object detection.

TABLE V. ANCHOR SIZES AND IOU ADJUSTMENTS EXPERIMENTS

Anchor Size & PIoU Setting	Average Precision			
	Small	*Medium*	*Large*	*All*
Baseline	39.729	65.876	82.137	51.873
Anchor-best[a] + PIoU-constant-best[b]	39.933	60.923	80.015	50.911
Anchor-best + PIoU-function-best[c]	**43.991**	65.127	86.632	**55.647**
Anchor-best	43.936	66.314	87.316	55.117
PIoU-constant-best	42.023	65.658	84.573	53.795
PIoU-function-best	40.794	62.152	85.335	51.447

a. Anchor-best has anchor size setting: [[2,4,8,16],[32],[64],[128],[256]].
b. PIoU-constant-best: $\gamma = 2.00$, $PIoU = IoU \cdot \gamma$ for all small objects.
c. PIoU-function-best: $PIoU = IoU \cdot f(x), f(x) = \min\left(\left\lfloor\frac{1024}{x}\right\rfloor, 1\right) \cdot 0.65 + 1$.

V. CONCLUSION

We investigated two methods for improving the performance of Faster R-CNN with the FPN model detecting small objects (area < 32^2) on a dataset in which targets are people with face masks. The first proposed method is adding smaller anchors (2^2,4^2,8^2,16^2) in the high-resolution feature map in FPN. The second proposed method boosts the IoU for small objects, which allows the model to propose more regions containing small objects. Through our experiments, we have verified that both methods effectively improve model prediction precision on small objects (face masks) detection. Adding smaller anchors in the low-level feature map in FPN shows that about a 10% increase in APs can be achieved. Boosting the IoU for small objects in the region proposal stage can increase about 5% APs. When adapting both modifications to the original architecture, we have seen a 10% increase in APs, with a 7.3% increase in average AP. The experiment results demonstrated that the proposed two methods could improve the model accuracy on detecting face masks, which are usually categorized as small objects in images. Furthermore, from the evaluation metrics for the models under different settings, we conclude that adding more anchors for detecting small objects in FPN shows a more obvious increase in detecting small face masks in the images than changing how IoU is calculated.

REFERENCES

[1] CDC, "Monitoring & Evaluation Action Guide: Community Mask Wearing," *Centers for Disease Control and Prevention*, June 22, 2021. https://www.cdc.gov/coronavirus/2019-ncov/php/mask-evaluation.html (accessed July 06, 2021).

[2] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, 2014, pp. 740–755.

[3] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image Vis. Comput.*, vol. 97, p. 103910, May 2020, doi: 10.1016/j.imavis.2020.103910.

[4] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," *ArXiv Prepr. ArXiv190207296*, 2019.

[5] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, and S. Yan, "Perceptual generative adversarial networks for small object

detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1222–1230.
[6] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem, "Sod-mtgan: Small object detection via multi-task generative adversarial network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 206–221.
[7] G. Chen *et al.*, "A survey of the four pillars for small object detection: Multiscale representation, contextual information, super-resolution, and region proposal," *IEEE Trans. Syst. Man Cybern. Syst.*, 2020.
[8] Y. Ren, C. Zhu, and S. Xiao, "Small object detection in optical remote sensing images via modified faster R-CNN," *Appl. Sci.*, vol. 8, no. 5, p. 813, 2018.
[9] "Face Mask Detection." https://kaggle.com/andrewmvd/face-mask-detection (accessed August 16, 2021).
[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *ArXiv150601497 Cs*, Jan. 2016, Accessed: June 19, 2021. [Online]. Available: http://arxiv.org/abs/1506.01497
[11] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression," *ArXiv190209630 Cs*, Apr. 2019, Accessed: August 15, 2021. [Online]. Available: http://arxiv.org/abs/1902.09630
[12] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 936–944. doi: 10.1109/CVPR.2017.106.
[13] B. Singh and L. S. Davis, "An analysis of scale invariance in object detection snip," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3578–3587.
[14] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." 2019. [Online]. Available: https://github.com/facebookresearch/detectron2