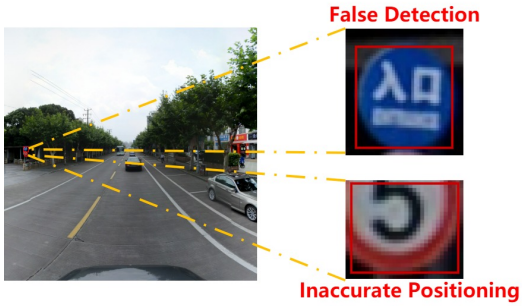
AN EFFICIENT FRAMEWORK FOR DETECTION AND RECOGNITION OF NUMERICAL TRAFFIC SIGNS

Zhishan Li[1,2], *Mingmu Chen*[2], *Yifan He*[✉,2,3], *Lei Xie*[1], *Hongye Su*[1]
[1]Zhejiang University, [2]Reconova Technologies, [3]Shenzhen Polytechnic

***Abstract*—Due to the variety of categories and uneven distribution of available samples, automatic traffic sign detection and recognition is still a challenging task. For those categories with less training data, existing deep learning methods cannot achieve desirable performance, and the overall detection effect is not satisfactory as well. In this letter, we fully explore the relationship between different traffic signs with digital characters and transform the category objects into multi-level classes to alleviate the uneven distribution of samples. We design a lightweight two-stage object detection framework with high real-time performance. The first stage network is proposed to obtain the category groups of traffic signs, and then we construct another object detection network to identify the digital characters of the detected traffic signs. To make the prediction in the first stage more accurate, we put forward a boxes fusion algorithm in the post-processing process and a refine module to improve the recognition performance. Experimental results show that our approach possesses significantly improved performance compared with the latest object detection networks and other traffic sign detectors. Even some traffic signs that only exist in testset can also be recognized accurately by our method.**

***Index Terms*—Traffic Sign Detection and Recognition, Real-Time, Digital Characters, Two-Stage.**

(a) Uneven distribution of different categories data. Each number in the second line represents the number of samples of the corresponding traffic sign in the training set.



(b) Common defects of existing models.

Fig. 1. Difficulities in detection and recognition of traffic signs with digital characters.

I. Introduction

AUTOMATIC detection and recognition of traffic signs with digital characters is important for Advanced Driving Assistance System (ADAS). With the progress of deep learning, object detectors based on convolutional neural networks, such as SSD[1], YoloV3[2], RetinaNet[3] and FasterRCNN[4] have achieved excellent performance in some object detection benchmarks [5][6][7][8]. However, accurate detection and recognition of traffic signs with digital characters is still a challenging task, as shown in Figure 1. Although the pattern of traffic signs is regular, the background is diverse and environmental factors such as illumination and weather conditions are complex. In addition, compared with other objects such as pedestrians or vehicles, most traffic signs are relatively small, which increases the difficulty of accurate positioning. Moreover, due to the wide variety of traffic signs, it is difficult to ensure that training data for each class is sufficient.

Methods based on machine learning have achieved great performance in traffic sign detection and recognition. Pei *et al.*[9] proposed a supervised low rank matrix recovery method in image sequence for traffic sign recognition. Tabernik *et al.*[10] improved the accuracy of traffic sign detection based on the improved Mask-RCNN[11] and data enhancement method. Serna *et al.*[12] proposed a multi-level network, which used Mask-RCNN[11] to detect and then utilized a small CNN to subdivide the categories. For digital types, there are some speed limit sign recognition methods based on machine learning. Miyata[13] extracted local binary pattern (LBP) features and recognized numbers using neural network to detect speed limit signs. Saadna *et al.*[14] proposed a cascade architecture of two linear support vector machines to detect the speed limit signs and used MNIST dataset to identify the digital characters. However, these speed limit sign recognition methods are unsatisfactory for other types of traffic signs with digital characters.

There are three aspects worthy of further consideration. Firstly, for those categories with less training data, the current traffic sign detectors cannot achieve desirable performance. Zhu *et al.*[15] proposed the TT100K dataset, and chose to ignore those categories with less than 100 data when calculating mAP. Wu *et al.*[16] proposed a real-time traffic sign detector based on YoloV3. For those categories with less data, Wu *et al.* proposed a data enhancement method to enhance the diversity of the training dataset. This optimization at the data level can indeed improve the fitting ability of the model. However, this is a little time-consuming and not suitable for traffic signs that only exist in the testset. Secondly, most traffic sign detectors[10][12][14] are cascade frameworks and directly combine the results obtained at each stage. The robustness of such a framework is relatively poor. If an error occurs at a certain stage, the whole result must be affected. At the same time, such a framework is easily affected by the positioning accuracy at the first stage, which has not been

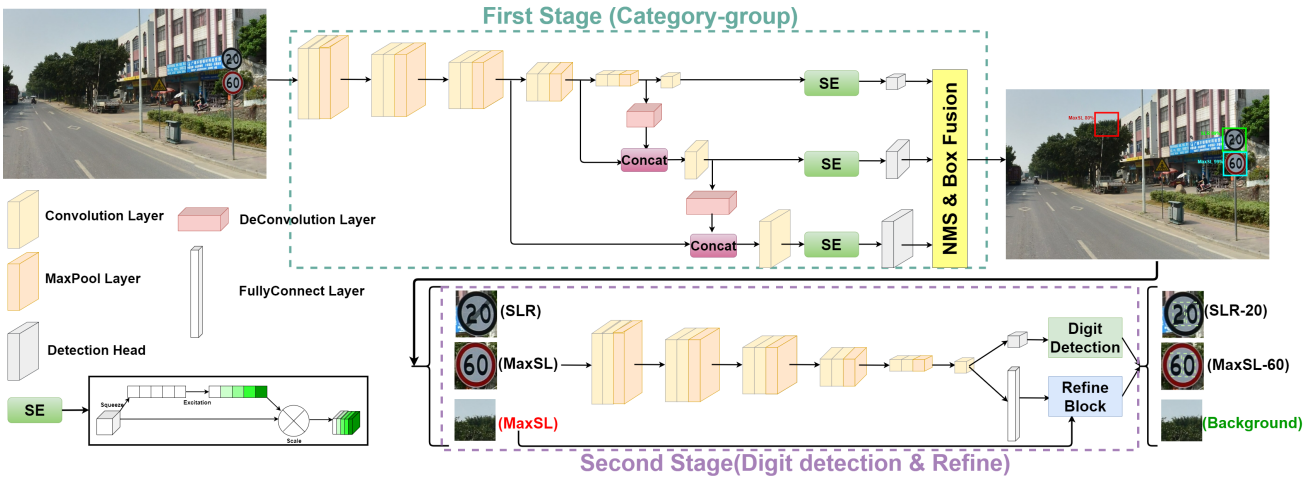


Fig. 2. Our framework for detection and recognition of traffic signs with digital characters.

considered and optimized. At last, most of the current traffic sign detection models[10][12][15] are based on large-scale convolutional neural networks, which require a large amount of calculation. This is not in line with the traffic scene with high real-time requirements.

For the above considerations, we propose different strategies to improve the performance.

For categories with insufficient training data, performance is undesirable if they are setting as independent classes[12][15]. After analyzing the relations between different types of traffic signs with digital characters, we propose a two-stage algorithm to effectively alleviate the problem of unevenly distributed samples. We design the first stage object detection network to detect and recognize the category groups of traffic signs with digital characters. In the TT100K dataset, the category groups are Height Limit (HL), Maximum Speed Limit (MaxSL), Minimum Speed Limit (MinSL), Speed Limit Release (SLR), and Weight Limit (WL). Then, different from the previous digit recognition methods in speed limit signs[13][14], we transform the classification task into the recognition of each digital character and propose an object detection network in the second stage. For instance, if an object detector possesses good recognition ability for MaxSL-100, it should also be able to accurately identify MaxSL-10. The reason is that their form is consistent and they have the same characters 1 and 0 inside. Through such a transformation, the distribution of samples is much more balanced than before.

For the robustness of the cascade framework of traffic sign detector, we propose a refine branch appended at the backbone of the second stage network. We combine the results of the refine module with the prediction results of the first stage network by weighted summation. Through the experimental results, we find that such a correction greatly increases the accuracy of category group prediction with little increase in calculations. In addition, for such a cascade architecture, the positioning accuracy of the first stage object detection is crucial for the recognition accuracy of the second stage network. In the post-processing stage, we borrow the idea of Weighted Boxes Fusion algorithm[17] and propose a more accurate boxes fusion method, which optimizes the original Non-Maximum Suppression(NMS) algorithm and improves the overall detection accuracy.

For real-time performance, we propose a lightweight two-stage object detection framework based on YoloV3. Besides, we insert Squeeze-and-Excitation[18] module to the backbone to improve the fitting ability of the model.

To sum up, the contribution of this article can be summarized as follows:

- We propose a lightweight two-stage algorithm to effectively alleviate the decline of accuracy caused by uneven sample distribution.
- We propose a refine module to correct the prediction errors of category groups in the first stage and enhance the robustness of the cascade framework.
- We propose a new boxes fusion post-processing algorithm to improve the positioning accuracy of the first stage network and overall performance.

II. Our Approach

A. The overall network structure

As we mentioned above, our overall framework composes of two lightweight object detection networks, as Figure 2 shows. The first stage network is used to detect five category groups. In the design of the network structure, YoloV3 is used as an initial architecture. In order to make the model lightweight, we replace the original backbone DarkNet53 with a lightweight VGG16 network whose kernel number of each convolution layer is half of the original VGG16[19]. In addition, we design a lightweight FPN structure to integrate features of different scales. Furthermore, SE block is used for feature recalibration to enable the network to automatically learn the importance of each channel from global information through backpropagation. Then, the effective features are enhanced and the invalid features are suppressed to improve the fitting ability of the network. We insert an SE block in front of the detector head of each branch to improve the performance of the first stage detector.

After detecting potential traffic signs, we design a lightweight object detection network to recognize the digital characters on them. Through such a transformation, we only need to identify 10 digital categories in this method. If we classify them directly, it will be far more than 10 classes. Because digital detection in the cutoff image is relatively simple, we design the second stage model based on Yolo[20], and the backbone is consistent with the first stage one for real-time performance. Our framework greatly alleviates the decline in accuracy caused by the uneven distribution of digital type traffic signs.

In addition, we propose a refine module based on the second stage object detection network. Due to the complexity and diversity of scenes, it is difficult for the first stage model to correctly predict the category groups in object detection. According to the cascade traffic sign detection framework, if the prediction in the first stage is wrong, the total result is false definitely. Our refine module is to modify the results of the first stage network to improve the accuracy of category groups prediction.

The refine block is a full connection layer connected to the second backbone. For each cutoff image, the classification result $\boldsymbol{P_1}$ in the first stage network is a vector containing the predicted score for each category group. We set that the output node of the full connection layer is also five so that the output $\boldsymbol{P_2}$ can be mapped to $\boldsymbol{P_1}$. Then, $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ are represented as:

$$\boldsymbol{P_n} = \{c_{n,1}, c_{n,2}, c_{n,3}, c_{n,4}, c_{n,5}\} \tag{1}$$

where $c_{n,j}$ represents the confidence of n stage network under each corresponding category group. We define each refine label according to the IoU between the position of cutoff object and Ground Truth (GT). If IoU is greater than 0.5, the refine label is the corresponding one-hot label. Otherwise, we define this object as background and set an all-zero vector as the label. Therefore, we use binary cross-entropy as the loss function L_{refine}, as shown in equation (2).

$$L_{refine} = \sum_{j=1}^{5}(-y_j \times log(c_{2,j}) - (1-y_j) \times log(1-c_{2,j})) \tag{2}$$

In the equation, y_j represents the j element of the label.

Finally, we combine the results of the first stage with the refine block through equation (3), and λ is the weight coefficient. Instead of using a direct concatenate strategy, we integrates $\boldsymbol{P_1}$ and $\boldsymbol{P_2}$ through weighted summation. As a result, $\boldsymbol{P_2}$ is the correction of $\boldsymbol{P_1}$. Because the refine module is the classification task on the cutoff traffic sign images, so the accuracy of category groups is significantly higher than that in the first stage. Such a correction greatly improves the performance of the overall framework. However, completely relying on $\boldsymbol{P_2}$ reduces the accuracy. We have verified in the experiment part.

$$\boldsymbol{P_{final}} = \lambda \times \boldsymbol{P_1} + (1-\lambda) \times \boldsymbol{P_2} \tag{3}$$

B. Boxes fusion algorithm

In our framework, the positioning accuracy of the boxes in the first stage greatly affects the accuracy of the second stage network and the performance of the overall framework.

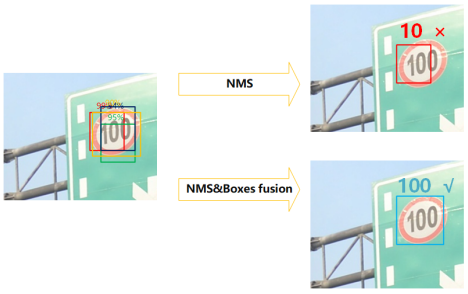


Fig. 3. Comparison of post-processing with/without boxes fusion.

As shown in Figure 3, there are four detection boxes around GT before post-processing. Due to the role of Non-maximum suppression (NMS), what we finally get is the box with the highest confidence. However, we cannot guarantee that this box is the most accurate one. Therefore, according to the original NMS algorithm, the cutoff object is recognized as 10, not 100.

Different from the original NMS, we do not directly take the box with the highest confidence as the final output box, but as an initial base box. For those boxes whose IoU is greater than 0.5 with the base box, we arrange them in descending order of confidence and select K boxes combining with the base box as candidate boxes. Then, we acquire the mean box of all candidate boxes according to equation (4). Finally, we obtain the fused final box by IoU weighting with the mean box, as shown in equation (5).

$$\boldsymbol{b_{mean}} = \frac{\sum_{i=1}^{K+1} \boldsymbol{b_{candidate,i}}}{K+1} \tag{4}$$

$$\boldsymbol{b_{final}} = \frac{\sum_{i=1}^{K+1} IoU_i \times \boldsymbol{b_{candidate,i}}}{\sum_{i=1}^{K+1} IoU_i} \tag{5}$$

In the above equations, $\boldsymbol{b_{candidate,i}}$ represents the coordinate vector of the i box in the candidate boxes set, $\boldsymbol{b_{mean}}$ represents the coordinate vector of the mean box, IoU_i represents the IoU of the i box with the mean box, and $\boldsymbol{b_{final}}$ represents the coordinate vector of the final fused box.

Solovyev *et al.*[17] proposed a method that fuses the box directly by confidence weighting. However, this method is based on the positive correlation between confidence and positioning accuracy. But in fact, they have no clear positive correlation. In our method, we do not directly take the box with the highest confidence as the initial base box, because this box is not the most accurate one in most cases. We obtain the base box by averaging the candidate boxes and compute the IoU of other candidate boxes with this mean box. Then, we set the IoU as weight to conduct boxes fusion. Experimental results show that our method achieves better performance.

III. Experimental Results

A. Experiment Settings

Dataset. TT100K is a large dataset of traffic signs. For the digital category, there are 61 classes in the dataset. We divide them into 3896 training data and 1961 test data. The input resolution is 640 $\times$ 640. The dataset of the second stage is based on the training set in TT100K. We extract the traffic signs according to GTs, and annotate the position and category of each digital character in the cropped numerical traffic signs.

Training Details. We first train the first stage network, and then the second stage model. In the training process, we set the batch size as 32 and the total epochs as 150. The learning rate of the first 80 epochs is 0.001, and 0.0001 for the remaining epochs. Parameters of the entire network are updated by Adam optimizer[21].

B. Results and Discussions

At first, we conduct ablation experiment about parameters λ, as shown in Table I. When λ is 1.0, the prediction of the category group is completely determined by the first stage model, and mAP of the total framework is 60.96. With the decrease of λ, the proportion of refine branch in the prediction results gradually increases, and mAP also gradually increases. Until λ is 0.2, the mAP of the overall frame reaches the maximum. However, when λ is 0.0, in other words, the category group results is completely determined by the refine branch, the overall mAP decreases slightly. This shows that the best performance is the two stages complement each other. Fully relying on the result of the refine branch is not the best method.

As shown in Table II, we intuitively show the mAP improvement brought by the optimization strategies. The mAP is only 29.57 when we directly use our first stage model to separately learn 61 classes of the dataset. After we combine the second stage object detection network, the mAP has been greatly improved to 57.35. Inserting SE block before the detection head of the first stage model brings an extra 3.61 improvement of mAP. Then, the refine branch of the second stage network significantly improved the mAP of the framework to 70.78. At last, we further improve the performance through the proposed boxes fusion algorithm. Compared with Weighted Boxes Fusion algorithm[17], our post-processing method achieves better performance.

We also compare our method with other detectors. Due to the uneven distribution of training data, the best mAP of the latest generic object detectors is 34.63. Besides, we re-implement four traffic sign detection algorithms and find that the method of Wu *et al.*[16] achieves 70.91 mAP. That is because this method produces sufficient training data through data enhancement, so as to improve the fitting ability of original model. For digital type traffic sign detector, we compare the extended speed limit detector of Saadna *et al.* and find that the detection performance of other classes of traffic signs with digital characters is unsatisfactory, Through the comparison with the above detectors, our method achieves the best performance in both inference speed and mAP. Furthermore, we migrate the data enhancement method of Wu *et al.*[16] and reach 78.95 mAP.

In order to show the effect of our approach more intuitively, we show the visualization results of several images after inference, as shown in Figure 4.

TABLE I
ABLATION EXPERIMENT OF λ

λ	0.0	0.2	0.4	0.6	0.8	1.0
mAP	69.01	**70.78**	67.99	65.31	63.63	60.96

TABLE II
COMPARISON WITH OTHER DETECTORS

Method	Input Size	mAP	FPS
Generic Object Detector:			
FasterRCNN[4]	1000×600	17.58	9
SSD[1]	512×512	21.62	21
RetinaNet[3]	800×600	30.47	8
YoloV3[2]	640×640	31.12	20
EfficientDet-D1[22]	640×640	33.25	29
YoloV4[23]	640×640	34.63	23
Traffic Sign Detector:			
Extended Speed Limit Detector[14]	1024×1024	27.59	21
MaskRCNN+Cat_CNN+Class_CNN[12]	1280×960	32.28	5
Zhu *et al.*[15]	1000×600	33.64	2
Wu *et al.*[16]	512×512	70.91	42
Our Method:			
First Stage only	640×640	29.57	97
Two Stage	640×640	57.35	82
Two Stage+SE Block	640×640	60.96	81
Two Stage+SE+Refine Block (BaseStruct)	640×640	70.78	80
BaseStruct+Our Boxes Fusion (BaseFrame)	640×640	**72.81**	**80**
BaseStruct+Weighted Boxes Fusion[17]	640×640	71.31	80
BaseFrame+Wu's Data Augmentation[16]	640×640	**78.95**	**80**

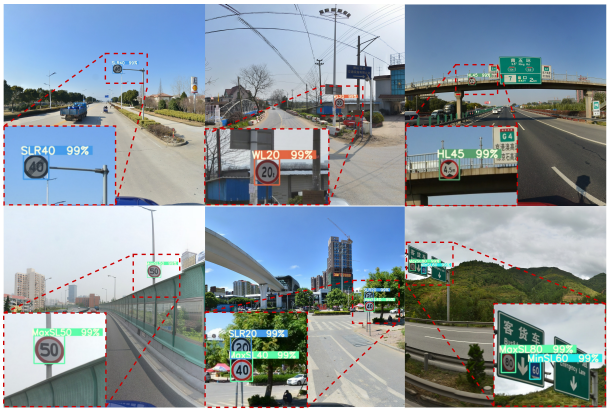


Fig. 4. Detection demo with our method.

IV. Conclusions

In this letter, we propose a lightweight two-stage object detection framework for detection and recognition of traffic signs with digital characters. By exploring the relationship between different traffic signs, our method effectively alleviates the impact of unevenly distributed samples. We further improve the detection performance by the proposed refine module and the boxes fusion algorithm. In the future, we will aim to propose a more general traffic sign detection algorithm to promote the development of ADAS and automatic driving.

REFERENCES

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[5] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan. 2015.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[7] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[8] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *arXiv preprint arXiv:1805.00123*, 2018.

[9] D. Pei, F. Sun, and H. Liu, "Supervised low-rank matrix recovery for traffic sign recognition in image sequences," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 241–244, 2013.

[10] D. Tabernik and D. Skočaj, "Deep learning for large-scale traffic-sign detection and recognition," *IEEE transactions on intelligent transportation systems*, vol. 21, no. 4, pp. 1427–1440, 2019.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[12] C. G. Serna and Y. Ruichek, "Traffic signs detection and classification for european urban environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4388–4399, 2019.

[13] S. Miyata, "Automatic recognition of speed limits on speed-limit signs by using machine learning," *Journal of Imaging*, vol. 3, no. 3, p. 25, 2017.

[14] Y. Saadna, A. Behloul, and S. Mezzoudj, "Speed limit sign detection and recognition system using svm and mnist datasets," *Neural Computing and Applications*, vol. 31, no. 9, pp. 5005–5015, 2019.

[15] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2110–2118.

[16] Y. Wu, Z. Li, Y. Chen, K. Nai, and J. Yuan, "Real-time traffic sign detection and classification towards real traffic scene," *Multimedia Tools and Applications*, vol. 79, no. 25, pp. 18 201–18 219, 2020.

[17] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, p. 104117, 2021.

[18] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[19] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[20] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 10 781–10 790.

[23] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.