ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

Research on the Use of YOLOv5 Object Detection Algorithm in Mask Wearing Recognition

Yifan Liu[1], BingHang Lu[2], Jingyu Peng[3], Zihao Zhang[4]

[1]University of California, Davis, Davis 95616, United States.

[2]School of NO.83, Xi'an 710043, China.

[3]School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Hubei 430074, China.

[4]Tongji University, Shanghai 200092, China.

Abstract

Masks can help people to reduce inhalation of droplets and the risk of infection. Because of the COVID-19, many governments required people to wear marks to prevent virus spread. In some public places, there are tons of people going back and forth everyday so it's impossible to settle a human monitor to identify whether everyone wears a mask. This work uses a different training version from YOLOv5 to train the dataset of mask wearing, and we use K-means to find the most appropriate anchors for datasets. Finally, by using data augmentation we get a more accurate model. Compared to human work, this model can be faster and more accurate to find a target and it can save countless money and time.

Keywords

Masks; YOLOv5; Computer vision; Machine Learning.

1. INTRODUCTION

Computer vision is appealing for use in automatic image analysis, detect events and automatic monitoring. It's a method that uses imaging systems to replace human's visual organs and uses computers to process and interpret images. Typically, computer vision includes three parts: the image processing, pattern recognition and image understanding. Many detection models based on computer vision are coming up, but it is very hard to detect minute objects and it's so hard to achieve real-time detection. From YOLOv1[1] to YOLOv5[2], people are always trying to come up with new models that can achieve all their hopes.

A previous work used Tensorflow to build a mask detection model. First, the work marked all images of the two categories in the datasets. By rotating and flipping every image in the datasets to achieve data augmentation. Then split the data into a train set, valid set and test set. After that use Flatten, Dropout and Dense to build a CNN model, and train CNN model. Then use the cascade classifier of Haar feature to detect face feature and through OpenCV to achieve face recognition. Finally, through OpenCV we can use the camera to detect masks. But this work has some drawbacks: the detection is not accurate and fast; it can't achieve real-time detection, and it's very hard to find mistakes and re-correct them.

In this work we chose a mask wearing datasets in roboflow and trained it by using a different version of YOLOv5. Then we chose three initial center points and calculated the distance between the rest points and these three center points; group together with the points with the shortest distance from the center; partition data points and recalculate center points to get the

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

Kmeans. Then use the Kmeans to find the most appropriate anchor for datasets. After that, by rotating and flipping every image in the datasets we achieve data augmentation. Then using data augmentation, we get a more accurate model.

2. METHOD

2.1. Development of YOLO Series and YOLOv5

With the development of technology, YOLO also iterates to become faster, stronger, and better. Now the YOLO series includes YOLOv1, YOLOv2, YOLOv3, YOLOv4 and YOLOv5. YOLOv1 is developed on the basis of R-CNN. It is known that R-CNN (Region Proposals + CNN) uses a convolutional neural network for target detection and, at the same time, add SVM (support vector machine) for prediction classification. Its position detection and object classification accuracy is very high. But the computation is heavy, and the detection speed is very slow.

YOLOv1 builds on R-CNN. Input images are processed only once (the origin of YOLO's name), different features are extracted through multiple convolutional layers, and convolution kernel parameters are shared each time. [3] The image detection speed is very fast. It is faster than the previous detection model and can meet the real-time requirements. It can reach 45 frames per second, using the full image as environmental information. However, the disadvantage is that the position detection accuracy is low and small objects cannot be detected.

YOLOv2 improves on YoLov1.The backbone network is upgraded. YOLOv2 uses average pooling, SoftMax classification and Anchor prediction box. Also, a combined training method of target classification and detection is proposed. Because of these improvements, the accuracy is improved obviously, especially for the detection of small objects, the disadvantage is that the detection accuracy of small objects is not high.

YOLOv3 makes some improvements. The convolutional layer of YOLOv3 is about 2.8 times that of YoLov2, increasing the depth and thickness of the network and thus increasing the model accuracy. SoftMax classifiers are replaced with multiple Logistic classifiers [4].

YOLOv4 appeared in 2019. The main purpose is to design a fast target detection system that can be applied in a real work environment and can be optimized in parallel. It uses data enhancement and some of the latest deep learning network tricks in recent years. Such as CutMix data enhancement, Swish, Mish activation functions [5].

YOLOv5 is the latest product in YOLO series. YOLOv5 is improved on the basis of YOLOv4, and its running speed is greatly improved, with the fastest speed reaching 140 frames per second. Meanwhile, the size of YOLOv5 is small, and the weight file is nearly 90% smaller than that of YOLOv4, which enables YOLOv5 to be deployed to embedded devices. Compared with YOLOv4, YOLOv5 has a higher accuracy rate and better ability to recognize small objects.

2.2. Use Kmeans to Update Anchor

Anchor boxes are used to predict bounding boxes. YOLOv2 starts with an anchor mechanism, increasing to 9 anchors in YOLOv3. According to the cluster VOC data set obtained by the author, there are 20 categories: small targets, like birds and cats, and big targets, like bicycles and cars. The size of the target varies greatly. The detection targets selected by individuals are inconsistent with those in YOLO, and some anchors cannot be rationally applied directly, so Kmeans is needed to calculate new anchors and improve the bounding box detection rate.

2.3. Data Augmentation of The Dataset

In deep learning, the number of samples is generally required to be sufficient. The more samples, the better the model effect and the stronger the model generalization ability. However, in practice, the sample quantity is insufficient or the sample quality is not good enough, so the sample data should be enhanced to improve the sample quality.[6] This work's target

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

applications may exist under various conditions, such as different directions, positions, proportions, etc. This work modifies the data using additional compositing and train a neural network to interpret these situations.

Methods for data enhancement:1.Data flipping; 2. Data rotation; 3. Image scaling; 4. Image clipping; 5. Image translation; 6. Add noise; The function of data enhancement: 1.Increase the amount of training data and improve the generalization ability of the model; 2.The noise data is added to improve the robustness of the model[2][6].

2.4. Common Target Detection Models

2.4.1 Faster R-CNN

Faster R-CNN is composed of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector [7]. that uses the proposed regions. RPN uses the base network to extract the features of the image using a series of convolution and pooling operations to obtain the original feature maps (gray areas), and then connects convolution layer and ReLU layer after the original feature maps to obtain what will be used to generate feature maps of region proposal. [8] Map each point in the feature maps back to the original image back to the center of the original image to get a reference point. According to the set scale size and aspect ratio, and around the reference point, k anchors are generated. Each point of the feature maps has output about k anchors, including whether there is a target, and the coordinate values of the k region proposal are returned.

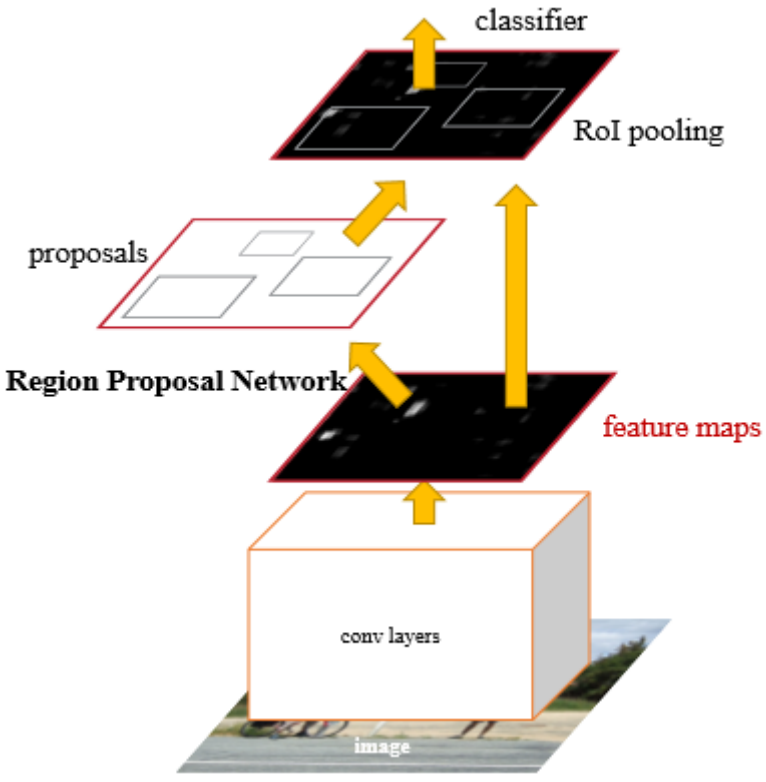

Fig 1. Faster R-CNN model[9]

2.4.2 SSD

The SSD method is based on a feedforward convolutional network, which generates a fixed-size bounding box set and the corresponding score of the target category in the box, and then generates the final detection result according to the non-maximization suppression step. [10]

Similar to faster R-CNN, SSD also proposed the concept of anchor. [8] The feature map of the convolution output, each point corresponds to the center point of an area of the original image.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

Using this point as the center, six anchors (called default boxes in SSD) with different width to height ratios and different sizes are constructed. Each anchor corresponds to 4 positional parameters (x, y, w, h) and 21 category probabilities (voc training set is 20 classification problems, with the addition of whether the anchor is a background, a total of 21 classifications).

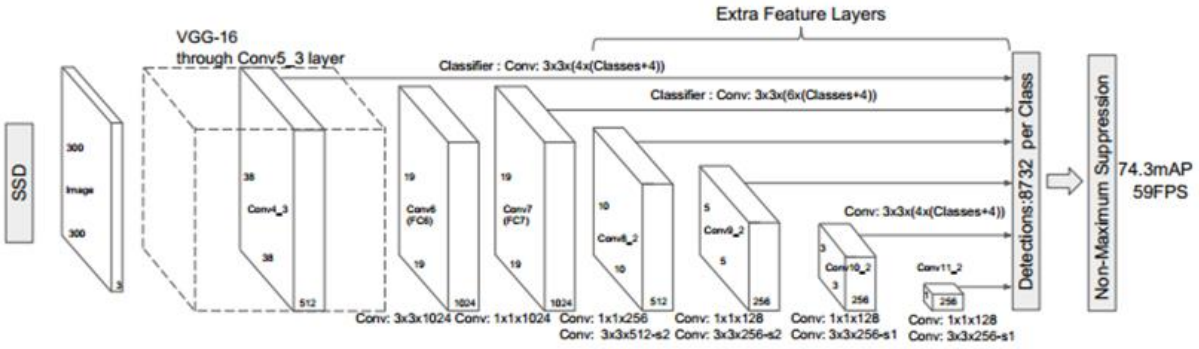

Fig 2. SSD network structure[11]

2.4.3 YOLO

The YOLO model is a standardized and real-time target detection. It is the most advanced real-time object detection system. It is another milestone target detection algorithm after RCNN, faster-RCNN, and SSD.YOLO V1 is based on darknet and is built into C. Darknet is an open source neural network framework written in C language and CUDA. [8] YOLO's real-time object detection technology for common usage problems solves the pain point in the detection --- speed problem and integrates target area prediction and target category judgment into a single neural network model. YOLO turns the detection process into a regression problem, simplifying many calculation processes.

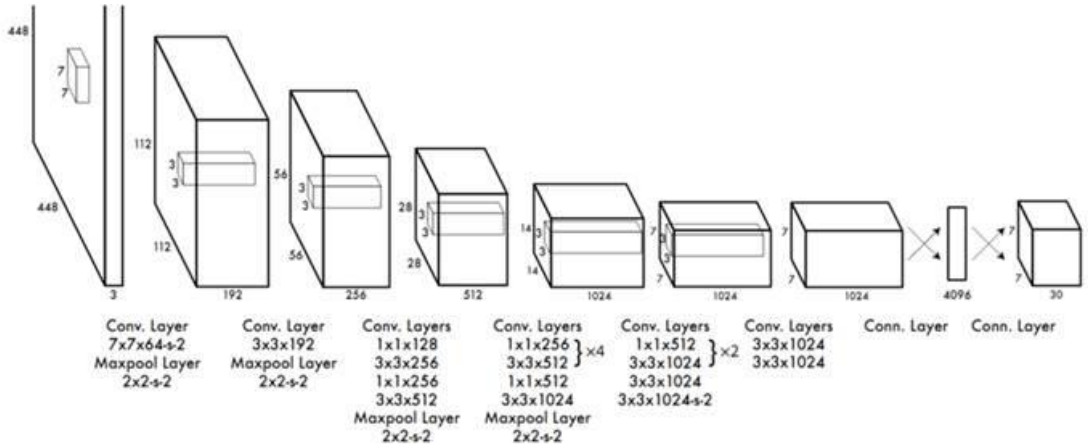

Fig 3. YOLO network structure [12]

YOLO can see the information of a whole image during training and testing, so YOLO can make good use of context information when detecting objects, so it is not easy to predict wrong object information on the background.

2.4.4 Comparison of three target detection models

Faster R-CNN has the best detection effect on small targets. The speed of SSD detection is the fastest, especially SSD mobile net. YOLO makes far fewer background mistakes than Fast R-CNN. By using YOLO to eliminate background detections from Fast R-CNN we get a significant boost

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

in performance.[11] YOLO v3 absorbs some of the advantages of the first two. It is faster than Faster R-CNN and smaller than SSD detection standards.

3. EXPERIMENTS

3.1. Dataset

We used a dataset provided by Robflow, after data augmentation, we have 447 images in total. And we divided these images into three parts, training, testing, and validation with a ratio of 7:2:1. The graph below shows the distribution of classes and central point of bounding boxes.

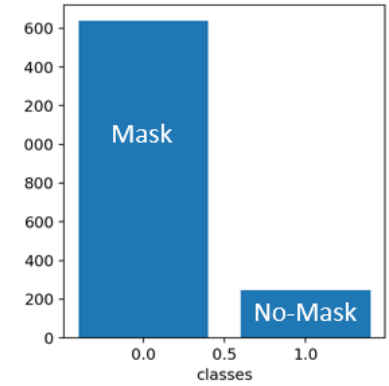

Fig 4. Distribution of classes. Produced by Yolov5

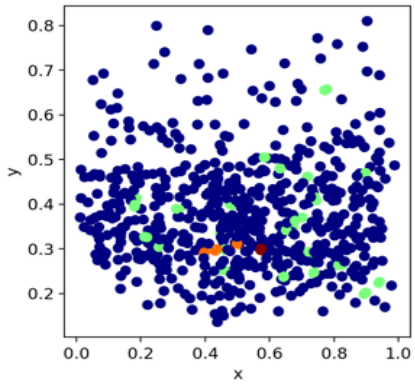

Fig 5. Distribution of bounding boxes' centers Produced by Yolov5

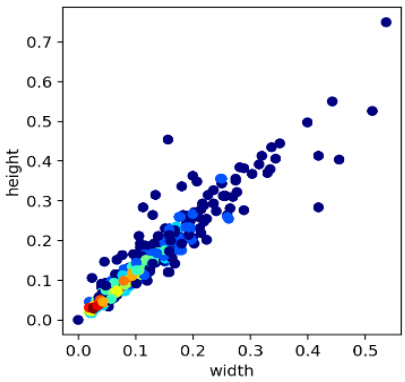

Fig 6. Distribution of bounding boxes's sizes Produced by Yolov5

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

Fig 7. Images with bounding boxes from our dataset. Produced by Roboflow [12]

3.2. Training Environment

The training process is done on the Google Colab online Python notebook tool. We have been assigned an Tesla P100 GPU with 16GB memory for our training. For the five different training methods provided by YOLOv5, time varies. Moreover, the size of each image in our dataset is 416, and we used batch size 16 for our training.

3.3. Find Appropriate Anchors

By using the label of each training image, we are able to use the K-means method to calculate the best anchors for our dataset. We keep the number of anchors the same as default but use the calculated anchors to redo the training we made in the last part.

3.4. Data Augmentation

For the data augmentation part, we used two ways. One is changing the brightness of images to 50% brighter or 50% darker, the other is the blurriness up to 1.5 times pixels for each image.

3.5. Data Collection and Analysis

During the training process we used the tensor board to trace and plot the losses for each session. And after the training is finished, we run the test program with the output weight on the test part from our dataset. Besides, we also tested those weights on videos which do not belong to the original dataset which could see if our trained models have limitations on our datasets.

4. RESULT

4.1. New Anchors

By using K-means to process the annotations of each image, we get new anchor values which is more appropriate compared with the default anchors provided by yolov5. Fig.8 shows the top 5 anchors of our dataset. Our new anchor boxes have a smaller ratio on height and width compared with the original anchor boxes, which is more suitable for vehicle recognition. The data accuracy is 68.54% which is higher than the accuracy with default anchors (67.07%).

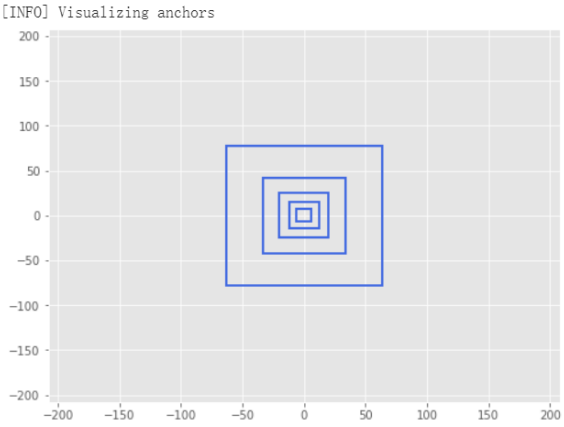

Fig 8. New anchor boxes for our dataset

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

4.2. Training Loose

Fig.9-13 shows the loss during training, different color means different training version. Orange for yolov5x, red for yolov5s, blue for yolov5m and indigo for yolov5s[12]. For the first three sessions, the training epochs are all 200. Our experiment shows that by adjusting the anchor, the training loss at the end point will drop compared with using default anchors. What's more, by using the data augmented dataset in the training process, the loss drops are more drastically shown by the slope. Finally, when we increase the training epochs to 400, the loss becomes more smooth during the training and approaches to 0.

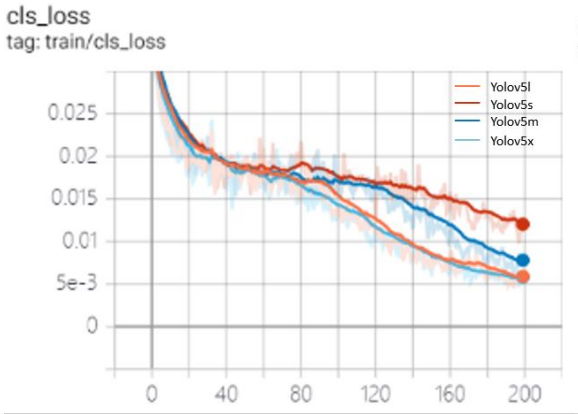

Fig 9. Loss with default method

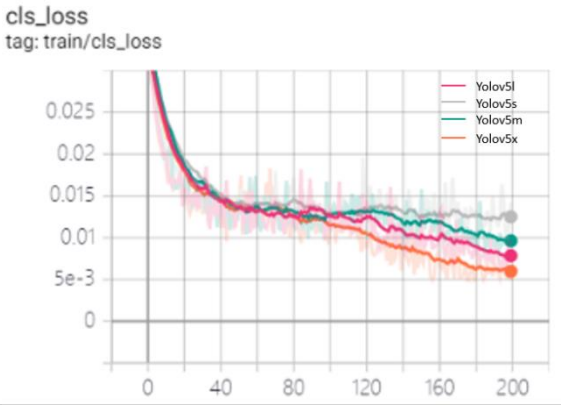

Fig 10. Loss after using new anchors

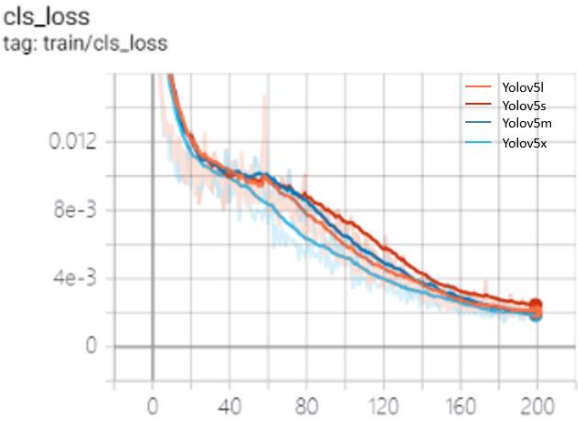

Fig 11. Lose after data augmentation

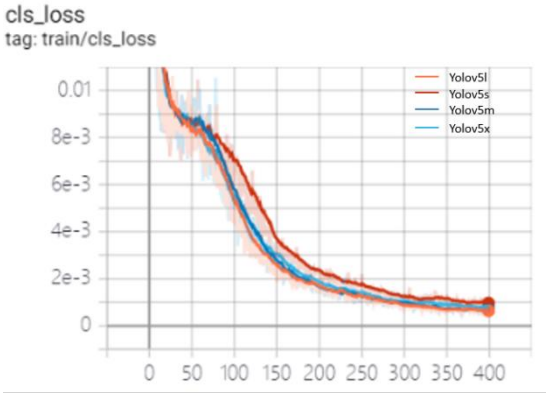

Fig 12. Lose after increasing epochs

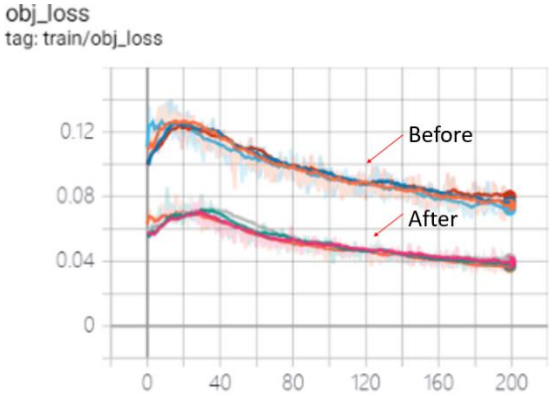

Fig 13. Lose comparison of changing anchors.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

4.3. Accuracy and Speed

Table.1 is the result of our trained model running on a computer without a Tesla-100 GPU. The anchor adjustments bring a considerably lower increase on the accuracy than the data augmentation and increasing epochs brought to the training process. And the processing time taken for each image with different training methods is almost the same except the last one which doubled the training epochs. Some of the output images were shown in Fig.14.

Table 1. Performance of each training method

Training Method	Precise Rate	Training Time	mPA	FPS
Default setting	31.7%	20min 58s	25.3%	100
Add anchor adjustment	36.2%	20min 48s	27.1%	100
Add data augmentation	78.5%	47min 22s	87.7%	90
Double epochs	90.2%	1h 37min 1s	92.1%	50

Fig 14. Our output running on the testing image set. Powered by YOLOv5.

5. CONCLUSION

Based on the YOLOv5 model provided by Ultralytics, we applied the anchor adjustment and data augmentation, which make the training process of our dataset more efficient and accurate than the default setting. With a 1h 37min training period, our output model could achieve an accurate rate of 92.1%. Our research shows that anchor adjustments and data augmentations could be used to improve the training efficiency with YOLOv5. With the feature of real-time video detection provided by YOLOv5, our model could do the real-time mask wearing recognition, which could be used in public spaces and warning people who do not wear a mask.

REFERENCES

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi: "You Only Look Once: Unified, Real-Time Object Detection", 2015; arXiv:1506.02640.

[2] Glenn Jocher, yolov5, (2020), GitHub repository, https://github.com/ultralytics/yolov5

[3] CSDN, Wanzhuandeepleaening, 2020, Theory and Evolution of the YOLOv4 Model, https://blog.csdn.net/shajiayu1/article/details/105755280

[4] Joseph Redmon, Ali Farhadi: "YOLOv3: An Incremental Improvement", 2018; arXiv:1804.02767.

[5] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao: "YOLOv4: Optimal Speed and Accuracy of Object Detection", 2020; arXiv:2004.10934.

ISSN: 2472-3703 DOI: 10.6911/WSRJ.202011_6(11).0038

[6] CSDN, Sanjingyesanjingye,2019,Deep Learning: Why Data Enhancement, https://blog.csdn.net/LEEANG121/article/details/102962798

[7] R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV), 2015.

[8] CSDN, Sankexin123, 2018, Detail the Three Most Commonly Used Models For Target Detection: Faster R-CNN, SSD and YOLO, https://blog.csdn.net/weixin_42273095/article/details/81699352

[9] Shaoqing Ren and Kaiming He and Ross Girshick and Jian Sun:" Faster R-CNN: Towards Real-Time Object Detection with Region Proposal.Networks", 2015; arXiv:1506.01497.

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan ,Christian Szegedy,Scott Reed, Cheng-Yang Fu ,Alexander C. Berg, "SSD: Single Shot MultiBox Detector," European Conference on Computer Vision (ECCV), 2016 (In press)

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan ,Christian Szegedy,Scott Reed, Cheng-Yang Fu ,Alexander C. Berg, "SSD: Single Shot MultiBox Detector," European Conference on Computer Vision (ECCV), 2016 (In press)

[12] Jacob Solawetz, Joseph Nelson. (2020) How to Train YOLOv5 on a Custom Dataset. https://blog.roboflow.ai/how-to-train-yolov5-on-a-custom-dataset/