ISSN 2616-7433 Vol. 2, Issue 11: 33-41, DOI: 10.25236/FSST.2020.021106

A Mask Detection System Based on Yolov3-Tiny

Guo Cheng[1], Shuyang Li[2], Yanheng Zhang[3], Ran Zhou[4]

1 School of Computing, Sichuan University, Chengdu 610065, China
2 The Henry Samueli School of Engineering, Irvine 92617, the United States
3 School of Mathematics and Information, Fujian Normal University, Fuzhou 350117, China
4 School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu 610000, China

ABSTRACT. *Currently, there is a global outbreak of novel coronavirus pneumonia, which infected many people. One of the most efficient ways to prevent infection is to wear a mask. Thus, mask detection, which essentially belongs to object detection is meaningful for the authorities to prevent and control the epidemic. After comparing different methods utilized in object detection and conducting relevant analysis, YOLO v3-tiny is proved to be suitable for real-time detection.*

KEYWORDS: *Object detection, Deep learning, Convolutional neural network, Yolov3-tiny*

1. Introduction

Object detection is to find all target objects in the image. It includes two subtasks: target localization and classification. In this case, the target objects are masks. Object Detection is one of the basic tasks in the field of computer vision, which has been studied for nearly 20 years. Recently, with the development of deep learning technology, it has changed from traditional algorithm to detection technology based on deep neural network. From the original R-CNN in 2013 to the M2Det in 2019, deep-learning based object detection technology has been applied to network architecture from two stages to one stage, from bottom-up only to top-down, from PC to mobile.

The two-stage algorithm requires generating proposal and then performing fine-grained object detection. Typical representatives of such algorithms include R-CNN, Fast R-CNN, etc. The one-stage algorithm extracts feature directly from the network, typically with YOLO. Next, we will discuss R-CNN series and YOLO series in more detail.

R-CNN (Region with CNN features) is a milestone leap in the application of convolutional neural network to object detection problems.[1] The algorithm can be seen in Fig. 1 and Fig 2.

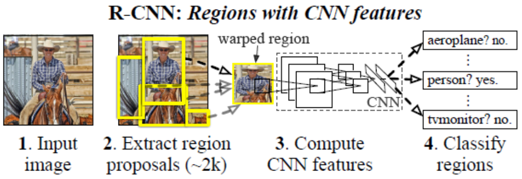


Fig.1 R-CNN Architecture [1]

(1) Generate candidate region proposals by using selective search.

(2) Process proposals by CNN first. Then, use SVM to classify the features.

(3) Use linear regression to generate a more precise bounding box.

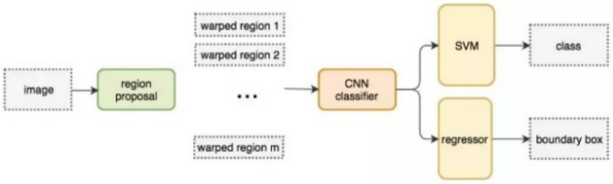


Fig.2 Flow Chart of r-CNN System

However, R-CNN suffers from severe speed bottlenecks since there is double computation. So, Fast R-CNN was born to solve the problem. The algorithm divided into the following steps (Fig.3 and Fig 4):

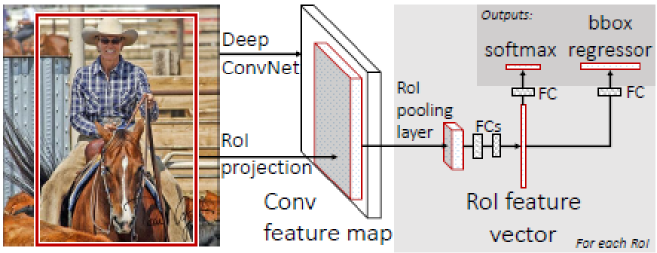


Fig.3 Fast r-CNN Architecture [2]

(1) The same as R-CNN.

(2) Input the original image to CNN with less calculation.

(3) Use ROI pooling to resize the candidate region proposals into one size.

Compared with R-CNN architecture, there are two main differences in Fast R-CNN: one is that the last convolution layer is followed by an ROI pooling layer, and the other is that the loss function uses a multi-task loss function to add bounding box regression directly to the CNN network training. This improvement can better preserve the features that are favorable for classification and regression.

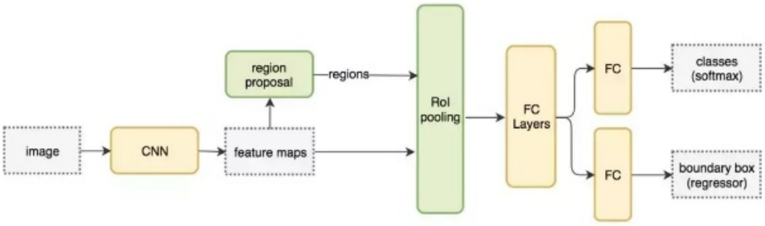


Fig.4 Flow Chart of Fast r-CNN System

Fast R-CNN relies on external candidate region methods such as SS, which runs on the CPU and is slow. Instead of using a specific algorithm to get candidate regions, it is a better choice to let the network learn what its candidate regions should be. Therefore, Faster R-CNN follows the same design as Fast R-CNN, except that it replaces the candidate area method with RPN (Fig. 5). It is more efficient in generating ROI and runs at a rate of 10 milliseconds per image.

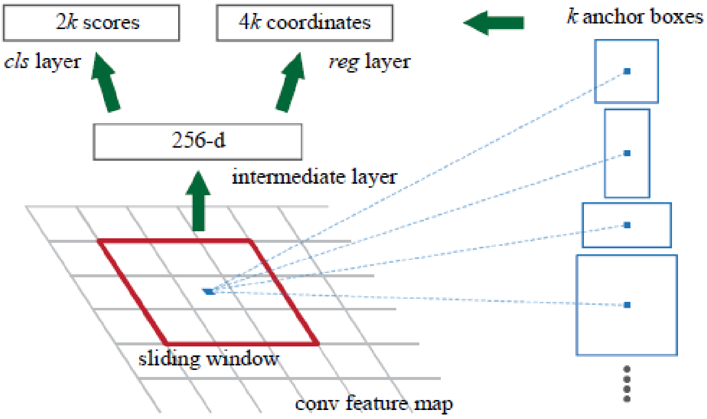


Fig.5 Region Proposal Network (RPN). [3]

Different from R-CNN series “look twice”, YOLO (You Only Look Once) combines the two stages of candidate area and object recognition into one. YOLOv1 was proposed in 2015. It regards object detection as a single regression task, using a single convolutional neural network to extract all the bounded boxes and class probabilities at a time (Fig. 6 and Fig. 7).

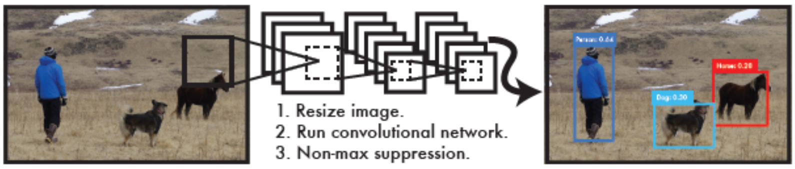


Fig.6 The Process of Yolov1. [4]

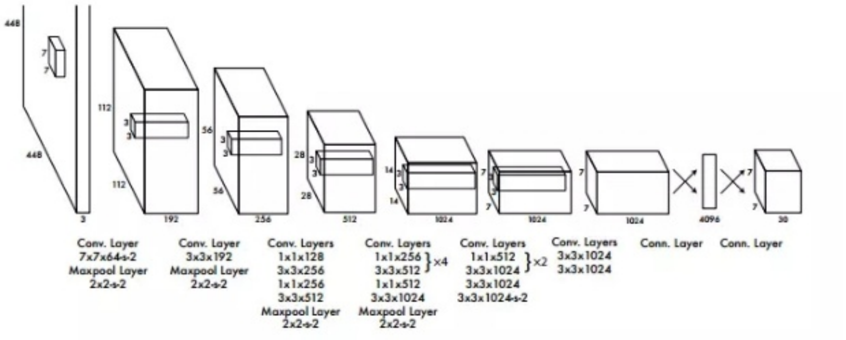
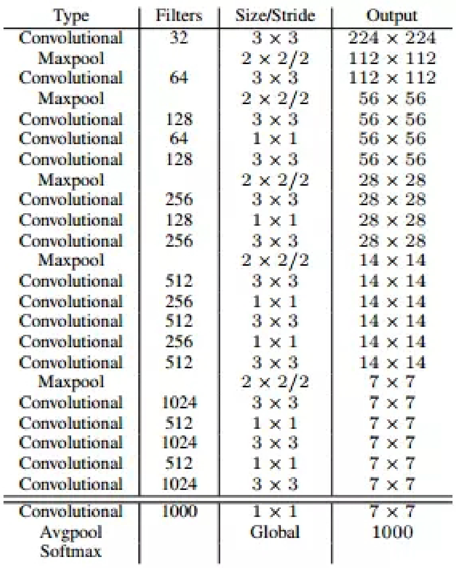


Fig.7 Yolov1 Architecture.

Inspired by GoogleNet, it uses only the first 20 convolution layers of the network and a full connection layer while being pre-trained on the ImageNet 1000 class. After pre-training, it is used in the detection network, adding 4 convolution layers and 2 full connections layers.

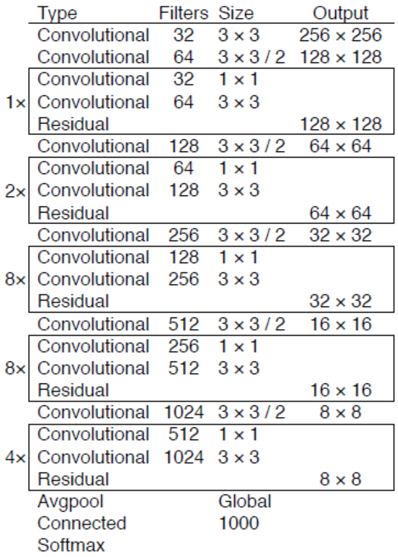
YOLOv2 is better, faster and stronger. Figure 8 shows Darknet-19 architecture. This network only contains 19 convolution layers and 5 max pooling layers. This is the key to reducing the amount of computation.



Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 2/2	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Convolutional	256	1 × 1	14 × 14
Convolutional	512	3 × 3	14 × 14
Maxpool		2 × 2/2	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	512	1 × 1	7 × 7
Convolutional	1024	3 × 3	7 × 7
Convolutional	1000	1 × 1	7 × 7
Avgpool		Global	1000
Softmax			

Fig.8 Darknet-19

On the basis of YOLOv2, YOLOv3 has the following improvements. First, it introduces the residual module and further deepens the network. The improved network has 53 convolution layers, named Darknet-53 (Fig. 9).



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Fig.9 Darknet-53. [6]

Second, Yolov3 has three outputs shown as figure 10.

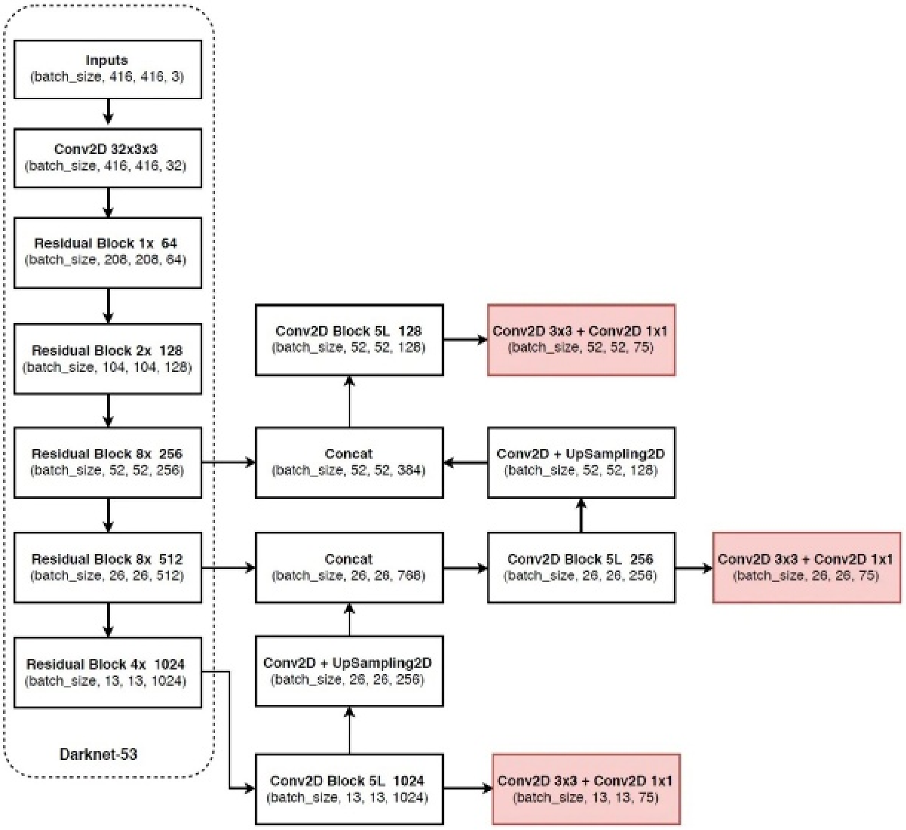


Fig.10 Yolov3 Architecture

Based on Yolov3, some feature layers have been removed and only two independent prediction branches have been retained. The structure is shown in Figure 11.

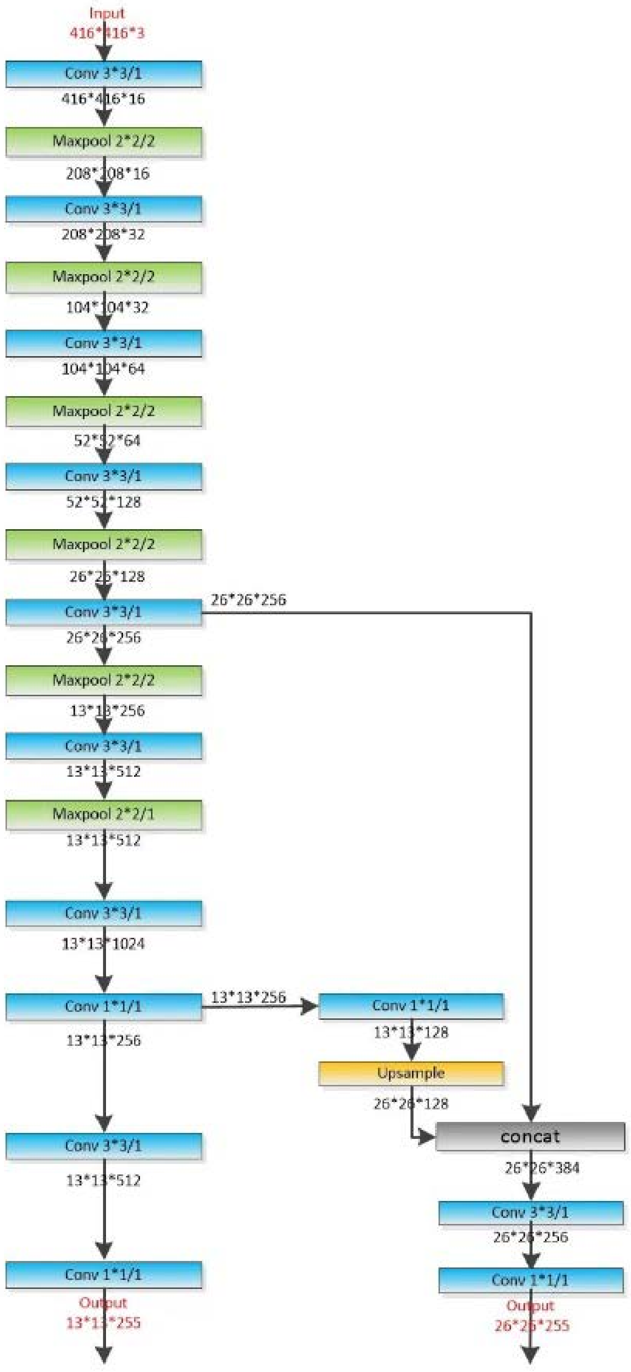


Fig.11 Yolov3-Tiny Architecture

2. Materials and Methods

2.1 Dataset

The dataset used in this paper consists of 234 front-face pictures of people without masks and 136 pictures of people wearing masks. Due to the limited images, data enhancement is needed. The yolov3-tiny model reads txt text to get the storage location of the training image, the pixel location of the target in the image and the category of the target. LabelImg is an image annotation tool that saves object class and location information labeled in an image as a file in xml format for model training. Figure 12 and 13 show the process of marking the image.

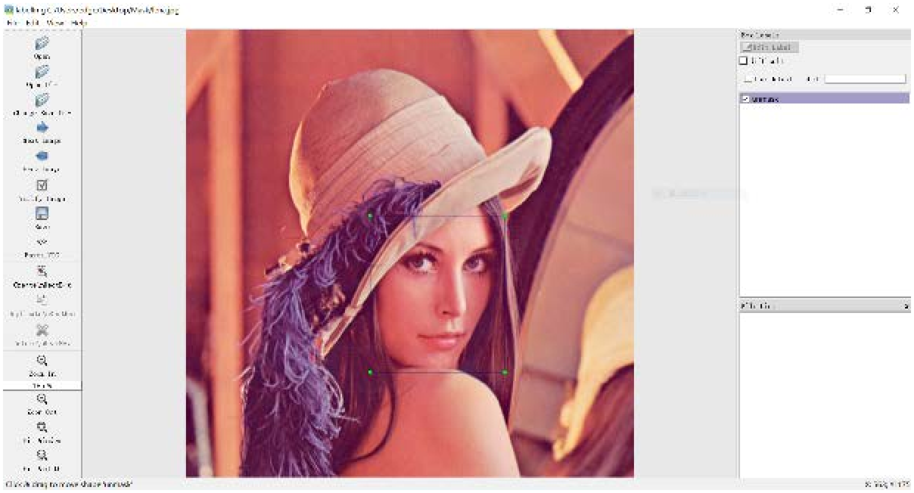


Fig.12 Labeling the Image of No Masks

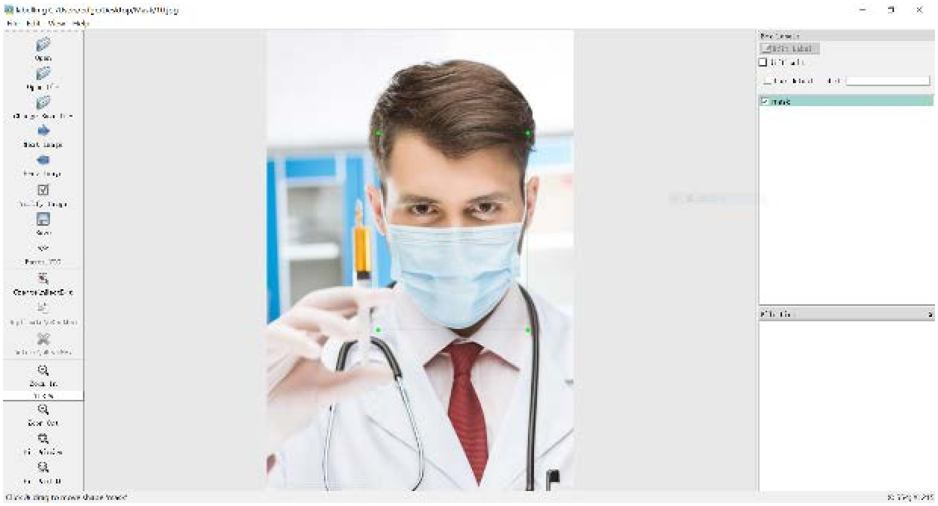


Fig.13 Labeling the Image of Wearing Masks

In this paper, the data is mainly enhanced by rotating the original images and the corresponding annotation files. By doing so, finally there were 468 pictures of people wearing masks and 544 pictures of people not wearing masks.

2.2 Experiment

The main performance indexes of the object detection model are detection accuracy and speed. In general, two-stage algorithm has an advantage in accuracy, while one-stage algorithm has an advantage in speed. YOLOv3-tiny model was utilized.

The experiment is divided into the following steps:

(1) Environment configuration.

(2) Download YOLOv3 from Yolo official website.

(3) Load dataset.

(4) Modify the model.

(5) Model training.

(6) Prediction.

The training method is to randomly select 90% images from the above data set as the training set and the remaining 10% as the testing set. The training set is trained for 100 epochs. The batch size is 16. The training optimizer uses the Adam optimization algorithm. The initial learning rate is 0.001 and the final learning rate is 0.0001.

Next, mAP and loss of the testing set are calculated, the weights models are saved every ten epochs, the model with the highest mAP and the lowest loss are also saved, named “best.pt”.

3. Results and Discussions

There are four kinds of binary detection results: TP (true positive), TN (true negative), FP (false positive) and FN (false negative).

Precision calculates the TP percentage of all retrieved items (TP+FP). (see equation 1)

$$precision = \frac{TP}{TP+FP} \qquad (1)$$

Recall calculates the percentage of TP in all relevant classes (TP+FN). (see equation 2)

$$recall = \frac{TP}{TP+FN} \qquad (2)$$

mAP (Mean Average Precision) is shown in equation (3):

$$mAP = \frac{1}{|Q|}\sum_{q \in Q} AP(q) \qquad (3)$$

The figure below shows the changes in each value over 100 epochs. It can be known that Precision and mAP fluctuate around 0.8.

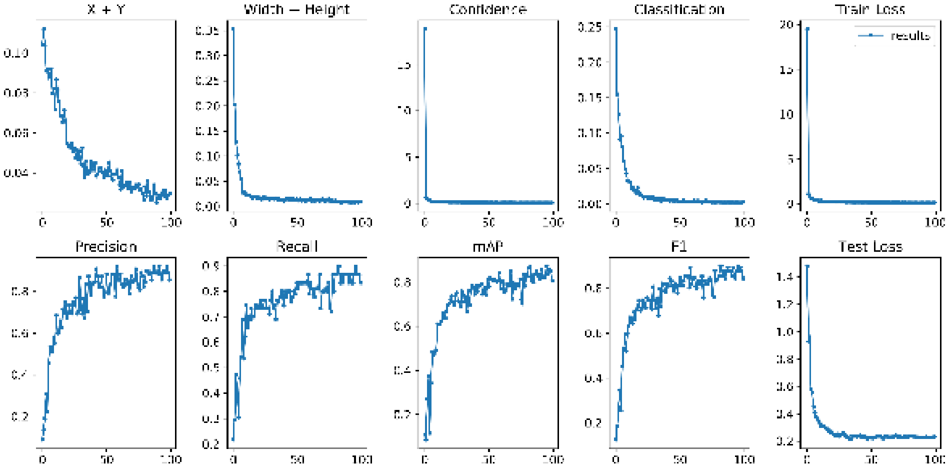


Fig.14 Changes in Values over 100 Epochs

From figure 15, it can be known that the system performs well in images of people’s front faces with or without masks.

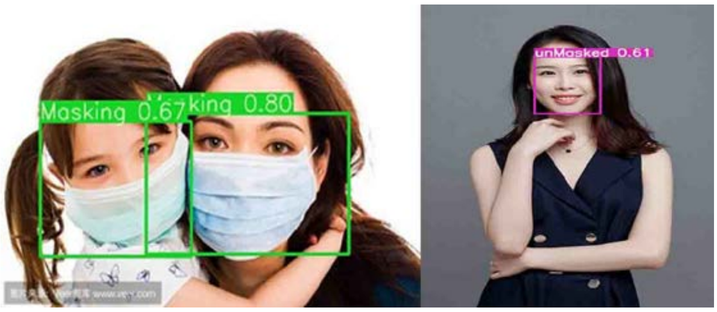


Fig.15 Clear Pictures of people's Front Face

In figure 16, the system still performs well.

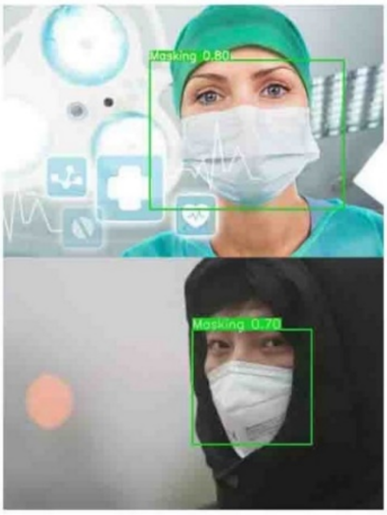


Fig.16 Front Face with Masks and Hats

In figure 17, for the kid on the left, the system did not work out because there is too much noise on the kid's face.

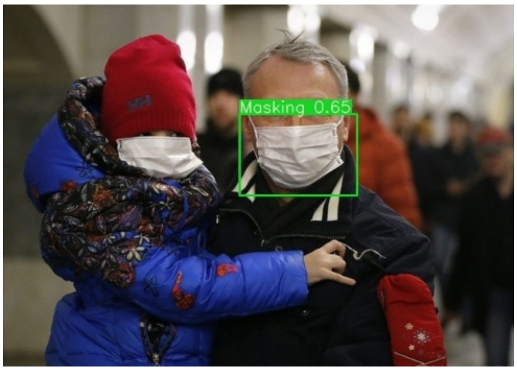


Fig.17 a Child and an Old Man

In figure 18, the model cannot tell the difference between a real person and a portrait sculpture. The left one cannot be predicted because of the light.

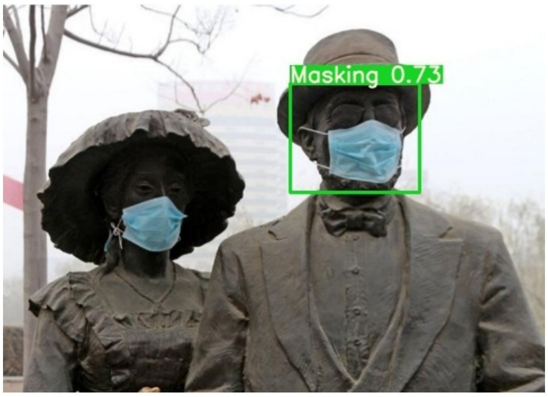


Fig.18 Two Sculptures

4. Conclusion

In this paper, a mask detection system is realized based on YOLOv3-tiny, which is small, fast and suitable for mobile hardware deployment and real-time detection. However, from the prediction, it shows very hard to quantify how much noise the system can handle and the result is not that accurate. Thus, the next step will focus on the accuracy of the detection.

References

[1] R. Girshick, J. Donahue, T. Darrell, J. Malik (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR.

[2] R. Girshick (2015). Fast R-CNN. IEEE International Conference on Computer Vision (ICCV).

[3] S. Ren, K. He, R. Girshick, J. Sun (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS.

[4] J. Redmon, S. Divvala, R. Girshick, A. Farhadi (2015). You only look once: Unified, real-time object detection. arXiv preprint rXiv:1506.02640

[5] J. Redmon and A. Farhadi (2017). Yolo9000: Better, faster, stronger. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 6517-6525. IEEE.

[6] J. Redmon, A. Farhadi (2018). YOLOv3: An incremental improvement. arXiv:1804.02767. .