Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 6, No. 3, 2023

Mask Detection Based on Yolov5s

Rongwei Zhang[1, *]

[1] College of Computer Science, Yangtze University, Jingzhou, 434025, China

[*] Corresponding author: Zhang Rongwei (Email: 1465569013@qq.com)

Abstract: Since the outbreak of the COVID-19 epidemic, wearing masks has become common sense and necessary protective equipment for go outside. The use of deep learning methods to detect whether a person is wearing a mask has also become a popular research direction in the field of computer vision. As an excellent object detection algorithm, Yolov5 is widely used in various fields. This article also applies the lightweight Yolov5s model for facial mask detection. Yolov5s uses a multi-scale detection method based on Feature Pyramid Network, which can effectively detect masks at different scales. This enables the model to obtain more accurate detection results on images of different scales. Yolov5s is a lightweight model with fewer parameters and faster detection speed compared to other Yolov5 models. The dataset in this article is from the Kaggle website. By preprocessing the dataset and training it on the Yolov5s network model, the trained model was tested and the effect of facial mask wearing detection was achieved.

Keywords: Deep learning, Facial mask detection, Yolov5s.

1. Introduction

The COVID-19 epidemic is a disaster all over the world. It is an acute respiratory infectious disease, and air transmission is the most common mode of transmission. Wearing masks correctly can effectively reduce the infection rate of the COVID-19 epidemic. It can be said that masks are our first guard against epidemic infection. During this period, many technology companies, such as Haikang and Baidu, produced mask wearing and temperature testing equipment, which improved the efficiency of protective personnel and reduced the pressure of protection. Traditional mask detection involves staff using the human eye to determine whether people are wearing masks, which wastes human resources and increases the burden on staff.

In this situation, it is very meaningful to use computer technology to improve the efficiency of mask wearing detection. Although the epidemic has been well controlled, it is still necessary to have a good habit of wearing masks when going out, especially in some crowded public places. By detecting pedestrians who have not worn masks and providing reminders, people's safety and health can be better guaranteed.

With the rapid development of computer technology and continuous integration with other fields, people's production and living standards have been significantly improved. Artificial intelligence, as one of the current and future hottest fields in the computer field, is widely applied in various directions. Applying the Yolov5s model to mask wearing detection not only completes the recognition task well, but also conforms to the trend of technological development. Liberating people from complex affairs is also a goal of technological development. Enable the model to autonomously learn the information in the data and use the trained model for mask wearing detection.

2. Related Theoretical Foundations

2.1. Convolutional Layer

The most common and basic network structures in deep learning are convolutional layers and pooling layers. The convolutional layer mainly uses convolutional operations. The process of convolutional operations is to use a certain size of convolutional kernel (which can be specified) to slide across the entire input image. During the sliding process, the product of the convolutional kernel and its corresponding position in the coverage area is calculated and summed, and then output.

The convolution operation is implemented by three basic units: input image (matrix), convolution kernel, and output matrix. The number of images, the height of images, the width of images, and the number of image channels are the four parameters of the input matrix. The parameters of convolutional kernels mainly include the size, sliding step, and number of convolutional kernels.

The two core ideas of convolutional layers are parameter sharing and local connectivity, both of which can reduce the parameters of the network. Parameter sharing refers to the use of the same set of convolutional kernels for operations during the sliding process of convolutional kernels. Local connection refers to the fact that nodes in a convolutional layer are only connected to some nodes in the previous convolutional layer, independent of other nodes in the previous layer, and learn local features. By repeatedly performing convolution operations on the extracted image, deep level information of the image can be extracted.

2.2. Pooling layer

The pooling layer is actually a type of downsampling. There are various forms of nonlinear pooling functions, among which "maximum pooling" is the most common. It divides the input image into several rectangular regions and outputs the maximum value for each sub region. This mechanism is effective because after discovering a feature, its precise position is far less important than its relative position with other features. The pooling layer continuously reduces the spatial size of the data, resulting in a decrease in the number of parameters and computational complexity, which to some extent also controls overfitting. Generally speaking, pooling layers are periodically inserted between the convolutional layers of deep learning.

The pooling layer has three main functions. First, it guarantees the feature invariance, including translation

invariance, Rotational invariance and scale invariance. Secondly, achieve feature dimensionality reduction and reduce the size of the input image, thereby reducing computational complexity. Finally, to prevent overfitting, overfitting hinders the model's generalization ability and performs very well on the training set used, but performs poorly on the new dataset.

2.3. Loss function and Activation function

The Loss function is equivalent to the evaluation between the real value and the predicted value. The Loss function is used to reduce the error between the real value and the predicted value. The Loss function must faithfully reduce all aspects of the model to a single number, and use the improvement of this number to measure the improvement of model performance. The choice of Loss function often determines whether the model can achieve the desired performance results. The Loss function provides the basis for the optimizer (back propagation).

The role of the Activation function is to introduce nonlinear factors to reduce the possibility of over fitting the model. Convolutional and pooling operations are all linear operations, but linear operations alone cannot handle complex and abstract data such as images and speech. By adding Activation function, each layer of the network can be nonlinear, thus improving the fitting ability of the model.

2.4. Fully connected layer

The fully connected layer serves as a classifier in the entire neural network, usually appearing in the last layer of the network, followed by the previous convolutional layer. In the neural network, the network before the full connection layer will map the original data to the high-level feature space, that is, feature extraction. The full connection layer is responsible for mapping the features learned from the network to the Sample space, playing the role of classification.

2.5. Yolov5s

There are many types of object detection algorithms in the Yolo series, among which Yolov5 holds a certain position in both academic research and practical applications, and is currently the mainstream object detection algorithm. There are four versions of Yolov5, namely Yolov5s, Yolov5m, Yolov5l, and Yolov5x. The main difference between these different versions is in the width and depth of the model. The wider and deeper the model, the larger the parameter quantity and the longer the detection time, but the accuracy of the model will also be more accurate.

Compared to the previous Yolo series algorithms, Yolov5s has undergone changes in both the input and prediction aspects. Firstly, in terms of data processing at the input end, in order to improve the detection performance of small target objects, Yolov5s and Yolov4 are consistent in using Mosaic data augmentation for the input images. However, Yolov5s also adopts an adaptive anchor box method to select the optimal anchor box value based on different sample sets during model training. In addition, for different sizes of images, the model also adopts an adaptive image scaling method at the input end during testing, which scales and fills the image with the least gray edges to improve detection speed. On the prediction end, Yolov5s adopts CIOU_ Loss and binary Cross entropy loss are calculated for frame loss, category probability and confidence loss respectively. At the same time, weighted NMS is used to cancel redundant prediction boxes. In addition, Yolo5s also adds optimization functions that can be selected according to needs, namely Adam and SGD (default). Adam is suitable for optimizing training smaller custom datasets, while SGD is used for optimizing training larger datasets.

3. Experiment

3.1. Experimental environment

Operating system	windows10 operating system
Graphics card	NVIDIA GeForce RTX 3080
video memory	12GB
CUDA Version	12.1
compiler	PyCharm
Programming Language	Python3.8

For 30 series graphics cards, if the CUDA version used is lower than 11.6, there may be running lag. It is recommended to use a higher version of CUDA.

3.2. Mask dataset

The main source of this dataset is the open source dataset provided through the Kaggle dataset website.

3.3. Experimentation

First, use anaconda to manage the required dependencies. The implementation of the Yolo model requires the use of many Python toolkits. To facilitate the management of these toolkits and their versions, it is recommended to use anaconda. Anaconda's virtual environment enables the configuration and installation of toolkits and versions in one environment to be isolated from other environments, greatly avoiding package conflicts, reducing the possibility of errors caused by toolkits, and reducing the pressure on developers. Switching between different environments is also very convenient, and now anaconda has become a very popular environmental management tool. After successfully installing the toolkit (including the corresponding version) required for the Yolov5s model, the model debugging phase can begin.

The coco dataset is a large and well-known dataset that can be used for object detection. Rotating 128 sheets from the coco dataset constitutes the coco128 dataset, which allows for model debugging. The structure of the dataset is divided into images and their corresponding image labels, using the Yolov5s model with a set number of training rounds of 100. The batch size is 16, and for each input image, regardless of its size, the image is uniformly converted to a size of 640*640. The optimizer selects SGD. The output results of the model are saved in the form of files, including the optimal weight file of the model, the weight file of the last training of the model, and the results of image annotation. After successfully debugging the model on the coco128 dataset, you can start training your own dataset.

There are many open source websites with rich datasets, and the dataset for this experiment mainly comes from the Kaggle website. After obtaining the dataset, it is necessary to process it accordingly to meet the standards of the data required by the model. After downloading, the dataset image data and label data are placed together, and the naming is also quite chaotic. Manual classification and renaming can be used to handle it, but it is very troublesome. Here, we use the method of writing Python code to solve the problem.

First, write the code for file classification, put the image

data together, label data together, and save the images in jpg format. The specific code logic is to sequentially read all files in the specified directory and move the file location according to the file suffix. After completing the file classification, divide the entire dataset (image data and label data) into training set, validation set, and testing set in a ratio of 8:1:1. Write code to rename the file, and test the training set, validation set, and testing set separately. The naming format is a number that starts from 1 and increases sequentially. The length of the number is the length of the training set's image data or label data, and if it is not enough, fill in the high order with 0.

After processing the dataset, write the corresponding yaml file for the dataset, which is used to specify the location of the dataset and the mapping of labels and categories. The Learning rate of the specified model is 0.01, the batch size is 16, and the number of training rounds is 100. The model is trained and the model is adjusted through the validation set in the training process. The training results and validation results are saved, but to determine the generalization of a model, it is necessary to test the test set. When testing the model, load the optimal weight file and test the test dataset.

3.4. Experimental result

The results of mask wearing test using Yolov5s showed that the precision value of Yolov5s was 0.84795, the recall value of Yolov5s was 0.86019, and the mAP_ 0.5 value of Yolov5s was 0. 87725, The value of mAP _0.5: 0.95 was 0.525.

From the detection results of the test set, it can be seen that the Yolov5s model performs well in detecting the majority of mask wearing scenarios, and performs well in multi target detection scenarios. Some pictures show people wearing masks without covering their noses, and Yolov5s also judges them as wearing masks. In fact, it can be more strictly judged, which is also a part that can be improved in the future. Some of the people in the pictures did not wear masks, but their mouths were blocked by their arms, which were also mistakenly detected as wearing masks.

4. Conclusion

This article uses the Yolov5s model to achieve mask wearing detection, which is of great significance for current epidemic prevention and control. The detection performance is excellent for single target and even multi target scenarios. The COVID-19 epidemic may be accompanied by humans for some time. For the epidemic, the best protective measure is to wear masks correctly. In some large shopping malls, there are usually dedicated testing personnel to ensure the safety of customers. The country and society are also actively guiding the masses to wear masks and enhance their awareness of protection. In some important places and densely populated areas, it is necessary to set up mask wearing detection points. Using the Yolov5s model for detection can reduce the pressure on staff and improve the efficiency of detection. In practical applications, providing richer and high-quality data to the model can improve its detection performance.

But some special scenarios, such as the mouth being blocked by a certain part of the body, often lead to false positives. Moreover, in multi-objective scenarios, some blurry characters are not detected, while some poster characters are also used as recognition targets for detection. These error cases are all directions that need to be further addressed and optimized in the future. The foundation for improvement is to first identify problems. Object detection has always been a popular research direction in deep learning. The continuous optimization and improvement of models have given them active popularity and also promoted the application of object detection in various fields. As an important model in the field of object detection, the Yolo model also has great development prospects. As a learner of deep learning, I will continue to learn and explore it.

References

[1] Liu Zhijia, Improved face mask detection algorithm based on YOLOX.Nanjing University of Posts and Telecommunications, 2022.

[2] Xu Dongdong Face mask detection and recognition based on deep learning. Jiangnan University, 2022.

[3] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition,2016: 770-778.

[4] Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines[J]. Microsoft Research Techinal Report, 1998, 10(1.43): 4376-4397.

[5] Howard A G, Zhu M, Chen B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv preprint arXiv:1704.04861, 2017.

[6] Huang G, Liu Z, Van Der Maaten L, et al. Densely connected convolutional networks[C]. proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition,2017: 4700-4708.

[7] Mercaldo F, Santone A. Transfer learning for mobile real-time face mask detection and localization[J]. Journal of the American Medical Informatics Association, 2021, 28(7): 1548-1554.

[8] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]. proceedings of the Proceedings of the IEEE conference on computer vision and pattern recognition,2014: 580-587.

[9] Sethi S, Kathuria M, Kaushik T. Face mask detection using deep learning: An approach to reduce risk of Coronavirus spread[J]. Journal of biomedical informatics, 2021, 120: 103848-103860.

[10] Wu P, Li H, Zeng N, et al. FMD-Yolo: An efficient face mask detection method for COVID-19 prevention and control in public[J]. Image and Vision Computing, 2022, 117: 104341-104351.

[11] Parkhi O M, Vedaldi A, Zisserman A. Deep face recognition[C]. British Machine Vision Conference,2015: 25-37.

[12] Wang C Y, Bochkovskiy A, Liao H Y M. Scaled-yolov4: Scaling cross stage partial network[C]//Proceedings of the IEEE/cvf conference on computer vision and pattern recognition. 2021: 13029-13038.

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

[14] H. Zhou, Z. Li, C. Ning, J. Tang, “Cad: Scale invariant framework for real-time object detection,” in The IEEE International Conference on Computer Vision (ICCV Workshop), 10 2017, pp. 760–768.

[15] Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. Arxiv 2020.05.

[16] Zhai Hongyu, Cheng Jian, Wang Mengyong. Rethink the IoU-based loss functions for bounding box regression. ITAIC 2020

- IEEE 9th Joint International Information Technology and Artificial Intelligence Conference, p 1522-1528, December 11, 2020.

[17] Xianwei Jiang, Bo Hu, Suresh Chandra Satapathy, Shui-Hua Wang, Yu-Dong Zhang, Chenxi Huang. Fingerspelling Identification for Chinese Sign Language via AlexNet-Based Transfer Learning and Adam Optimizer [J]. Scientific Programming, 2020, 2020.

[18] Song Z, Nguyen K, Nguyen T, et al. Spartan Face Mask Detection and Facial Recognition System[C]. proceedings of the Healthcare, Multidisciplinary Digital Publishing Institute, 2022: 87-111.

[19] Mata B U. Face Mask Detection Using Convolutional Neural Network[J]. Journal of Natural Remedies, 2021, 21(12 (1)): 14-19.

[20] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]. proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence,2020: 12993-13000.

[21] Liu L, Ouyang W, Wang X, et al. Deep learning for generic object detection: A survey[J]. International journal of computer vision, 2020, 128(2): 261-318.

[22] Shankar K, Lakshmanaprabu S K, D Gupta, et al. Optimal feature-based multi-kernel SVM approach for thyroid disease classification[J]. The Journal of Supercomputing, 2020, 76(28):1-16.

[23] Qin B, Li D. Identifying facemask-wearing condition using image super-resolution with classification network to prevent COVID-19[J]. Sensors, 2020, 20(18): 23-26.

[24] Inamdar M, Mehendale N. Real-time face mask identification using facemasknet deep learning network[J]. Available at SSRN, 2020,30(5):55-56.

[25] Yadav S. Deep learning based safe social distancing and face mask detection in public areas for covid-19 safety guidelines adherence[J]. International Journal for Research in Applied Science and Engineering Technology, 2020, 8(7): 1368-1375