北京郵電大學
Beijing University of Posts and Telecommunications

Queen Mary
University of London

Undergraduate Project Report
2021/22

Facial Mask Wearing Identification System Based on Deep Learning

Name:	Yongqi Yang
School:	International School
Class:	2018215108
QMUL Student No.:	190015176
BUPT Student No.:	2018212807
Programme:	Telecommunications Engineering with Management

Date: 28-04-2022

Table of Contents

Abstract

Since the outbreak of COVID-19 pandemic, people have suffered from its high transmissibility and fatality rate. It has been proved the wearing a facial mask can effectively prevent it. To reduce the cost of reminding people to wear masks, the author introduced an identification system based on deep learning, which has the capability to detect whether people wearing a mask or not.

In this paper, the system proposed by the authors has three detection modes: image detection, video detection and real-time detection. Users can choose the mode by themselves. The core of the system is a two-stage mask detector. The first stage is a Multi-Task Cascaded Convolutional Networks (MTCNN) based face detector. It can detect and extract all faces in the image for later classification. For second stage, the author proposes a CNN with 3 convolutional layers and 2 linear layers. Based on that, a face mask detector is trained. This project proposes a dataset with 100,000 images, 15,000 face images wearing masks. The mask classifier is trained and tested using this dataset achieves 93.15% accuracy and 94.6% recall. And the total model size is 8.8m. The performance meets the actual use requirements while ensuring light weight.

Keywords: deep learning, face detection, mask detection, MTCNN

摘要

自新型冠状病毒疫情爆发以来，人们饱受其高传染性和致死率之苦。事实证明，戴上口罩可以有效保护自己。安排专人提醒人们佩戴口罩是一个耗时耗力的工作。为了有效解决这一问题，作者引入了基于深度学习的口罩识别系统，该系统能够判断人们在公共场所是否戴口罩。

在本文中，作者提出的系统具有三个检测模式：图片检测，视频检测和实时检测。用户可以自行选择模式。系统的核心是一个两段式的口罩检测器。第一阶段是基于 Multi-Task Cascaded Convolutional Networks（MTCNN）的人脸检测器，它可以检测并提取图像中的所有人脸用于之后的分类。 对于第二阶段，作者提出了一个具有 3 个卷积层和 2 个线性层的卷积神经网络。在此基础上，作者训练了一个面罩分类器。它可以将人脸分类成佩戴口罩和未佩戴口罩，并将他们标注出来。该项目还提出了一个具有 100,000 张图片，其中 15,000 张佩戴口罩的人脸图片的数据集。使用这个数据集训练得到的口罩分类器的准确率达到了 93.15%，召回率达到 94.6%，而且总的模型大小为 8.8m。性能满足实际使用需求的同时保证了轻量化。

Chapter 1: Introduction

COVID-19 coronavirus epidemic has spread all over the world. Wearing face mask is an effective way to protect people. Arranged a staff to remind people to wear masks at the door is costly and inefficient. To overcome this situation, the author implements a software with a robust facial mask detector using deep learning to decrease the cost. The main function of the software is to monitor whether a person is wearing a mask or not. In this paper, the author applies deep learning technology to face detection, and traditional image processing technology is combined to identify the state of face wearing masks, so as to solve the daily needs of oral sense wearing detection, effectively improve detection efficiency and avoid the risk of cross contamination.

At present, mask testing is already a very hot field. Many companies, including Baidu and Huawei, have already commercialized it. There are also many scholars who have done research on mask testing. Shay.E [1] proposed a face mask detection robot based on RetinaNet, MTCNN and MobileNet v2, which can achieve F1 score of 87.7% with a recall of 99.2% in a variety of situations. Susanto [2], et al., implemented a face mask detection system based on YOLOv4, which can detect face mask with average 11 FPS. Vinh [3], et al., proposed one detector with accuracy of 90.1% based on YOLOv3 algorithm. However, these existing methods use large number of parameters and high requirement of hardware. And some of them are trained and tested on a small dataset, can't get a widely used result. So, in this paper, the author proposes a lightweight face mask detector that can be more portable and equivalent.

To build this face mask, the author chooses computer vision technology based on deep learning. The convolutional network structure of face detection was optimized and redesigned to meet the requirements of practical use. In this paper, the author utilizes the Multi-Task cascaded Convolutional Neural Network (MTCNN), which is one of the most popular face detection and alignment technique. The model uses a lightweight CNN with good accuracy and meeting the requirement of practical use.

What makes this paper different is that to achieve better practicability and portability, the system proposed in this paper constructs a CNN network with fewer layers. This sacrifices a bit of accuracy but minimizes the size of the model. The final accuracy and recall rate can also meet the basic use requirements. In addition, this paper uses open-source software

MaskTheFace[4] to add different kinds of simulated masks to the face data set LFW. This was combined with images from RMFD to build a dataset of over 100,000 images. This makes the data set for the entire project more robust and diverse, and also makes the test results more convincing.

The rest of the paper is organized as follows. In Section 2, the author discusses the background of the system, including social background, related technology including face detection technology, Faster R-CNN, YOLO and MTCNN. Section 3 describes the design and implementation of the face mask detection system. In this part, the author introduces the workflow of the system and face mask detector, shows design and training of face detection model based on MTCNN, and a self-construct neural network face mask image classifier. Section 4 shows the result of system and analyze the performance. And finally, the author draws a conclusion on my project and the future of it in Section 5.

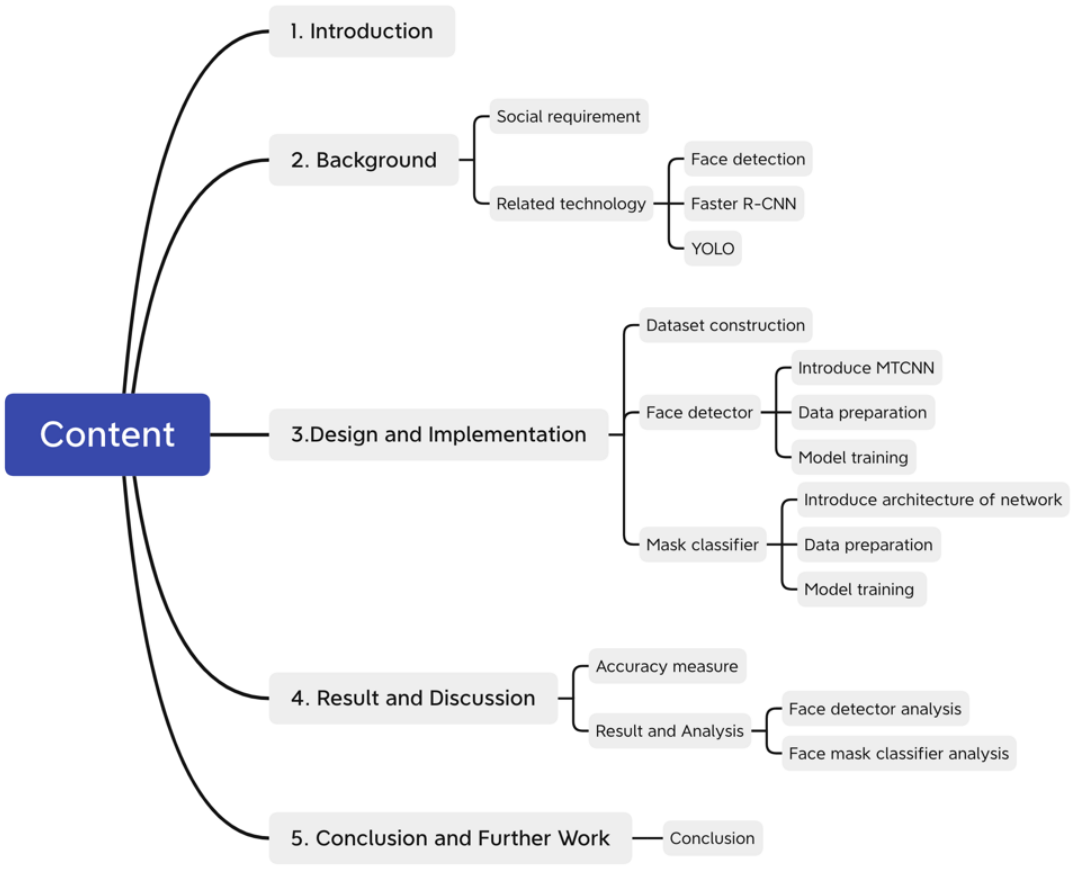


Figure 1. Content of this paper

Chapter 2: Background

2.1 Requirement

As of 5 April 2022, there have been more than 49 million confirmed cases of COIVD-19, including 6 million deaths, reported to WHO [5]. The known transmission routes of the COVID-19 include respiratory transmission and contact transmission. Respiratory transmission includes droplets or aerosols with pathogens, and contact transmission means that the virus attaches to the contaminated skin and enters the human body through mucous membranes. Preventing and cutting off the transmission route is the most effective key step to prevent the spread of the virus. And many governments have forced people wearing mask correctly in public areas.

Taking Beijing, China as an example, at the entrance of most public places, an average of two staff members are assigned to check whether visitors are wearing masks. It can be seen from this that the importance of wearing a face mask in public is beyond doubt. How to efficiently and accurately detect whether pedestrians wear masks in public has become an important research task, and the problem of mask wearing detection has been raised. In public places such as train stations and large shopping malls, relying on manual methods to carry out the wear test of the mouth slip not only increases the labor cost, but also increases the risk of the stuff being infected by the virus. Enlightened by this, an automatically face mask detector based on deep learning comes to my mind.

2.2 Related Work

2.2.1 Face detection algorithm

Face detection is an important part of face recognition task. Therefore, it has always been a hot point that how to obtain high-precision face region. The development process of face detection algorithm can be divided into three stages in chronological order, which are traditional template matching algorithm, statistical model for face detection and face detection method based on deep learning. In this paper, the author mainly uses the deep learning method.

In recent years, deep learning has been widely applied to other problems in the field of computer vision by researchers due to its excellent performance in the field of image classification. At the same time, face detection using deep learning algorithms has been widely studied and a

series of solutions have been proposed. The research shows that the key problem in the development of face detection technology in traditional unconstrained environments is that features such as Haar and HOG cannot obtain salient information of faces under various poses and lighting conditions. This limitation not only reduces the accuracy of face detection, but also reflects the difficulty of manually extracting complex features. Face as a special research object in the field of target detection, combined with the deep learning algorithm, the detection effect has been greatly improved, so the face detection algorithm framework based on deep learning has been widely proposed. For example, the target detection multi-classification network uses the face data set to train to obtain the detection model of face two-classification. At the same time, with the continuous introduction of new frameworks of deep learning, the combination of deep learning and traditional face detection algorithms has also designed a cascaded face detection network model with excellent performance. Face detection algorithms based on deep learning are mainly divided into three categories, namely cascade face detection algorithms, one-stage face detection algorithms and two-stage face detection algorithms.

2.2.2 Faster R-CNN

Faster R-CNN [6] is one of the best two-stage object detection neural network. Based on Fast R-CNN, it proposes an optimization scheme. Using a fully convolutional network called Region Proposal Network (RPN), it can simultaneously predict the object bound and the score of the object at each location. Using RPN, high-quality region proposals can be generated through end-to-end training. The Faster R-CNN algorithm is formed by integrating RPN and Fast R-CNN. Similar to the mechanism of attention, RPN can guide the algorithm where to look.

The Faster R-CNN network is mainly divided into four parts. Firstly, the conv layers, which is used for feature extraction. a feature through backbone after resizing the input image. Secondly, the RPN network which is the substitution of selective search of former RCNN [7] will generate the candidate boxes. There are two masks in this layer: determine whether all preset anchors belong to positive or negative, and fix anchors to get more accurate proposals. Thirdly, the ROI layer is used to collect the proposals generated by RPN, as well as extract them from the feature maps in CONV layers to send them to the subsequent full connection layer for classification and regression. Finally, the specific categories are calculated by means of proposals feature maps, and the exact final position of the detection frame is obtained by a bounding box regression.

The emergence of the Faster R-CNN model has basically perfected the detection algorithm system based on the candidate frame. Although its detection performance has been improved through continuous updates, it still cannot fully meet the timeliness requirements. At the same time, a simple and fast target detection model has become the trend of development, so that the regression-based detection model should be born.

2.2.3 YOLO

The YOLO[8] model defines the detection process as a regression problem and simplifies the detection and recognition process of objects by the visual neural network, showing excellent timeliness. The product network regression can realize the fast detection of objects. YOLO v3 is an outstanding representative of "One-stage" and is also one of the classic models of target detection.

The YOLO v3 algorithm uses Darknet-53 as a backbone for feature detection. It consists of 106 layers, including convolutional layers, residual layers, and up-sampling layers. The Darknet-53 network replaces the pooling operation by increasing the step size in the convolution operation to reduce the loss of low-level features and adds a batch normalization layer and an activation function layer after each convolution layer to speed up the convergence speed, prevent overfitting and other problems. The core idea of YOLO V3 is to partition the raw image with three different grids. The 13*13 grid divides each piece into the largest, used to predict large objects. A 26*26 grid divides each piece of medium size for predicting medium objects. The 52*52 grid divides each piece into the smallest and is used to predict small objects.

2.2.4 MTCNN

In 2016, the Multi-Task Cascaded Convolutional Network (MTCNN) [9] proposed by Kaipeng Zhang et al. has been well improved for the problems found in previous research. A novel lightweight cascaded CNN-based joint face detection and alignment framework and an efficient online hard sample mining method are proposed. The model mainly adopts three cascaded networks and adopts the idea of candidate box and classifier to perform fast and efficient face detection. The three cascaded networks are PNet for quickly generating candidate windows, R-Net for filtering and selecting high-precision candidate windows, and O-Net for generating final bounding boxes and face key points. And many convolutional neural network models that deal with image problems, this model also uses feature pyramid networks (FPR), bounding

regression, non-maximum suppression (NMS) and other techniques.

In the detection of MTCNN, the image to be detected is scaled by a certain proportion to adapt to faces of various scales. The essence of face detection is to multiply the images in the candidate area with the template according to a certain weight. When the size of the detected image is consistent with the size of the template image, the detection accuracy can be increased. Therefore, before using the P-Net network to detect the image, the image is reduced according to a certain ratio until the side length of the input image is equal to the preset minimum size, thus forming an image pyramid, obtain images of various sizes to be detected. The formula is showed in Equation 1.

$$dst = src * \frac{12}{minsize} * factor^n, n \in \{0,1,2,3,\cdots,n\} \quad (1)$$

***src* is the original image, *minsize* is the minimum detectable size, and *factor* is the scaling factor of the pyramid.**

(1) P-Net is the first convolutional network of MTCNN. The purpose is to quickly generate candidate boxes and boundary vectors. The input is a fixed size 12 × 12 × 3 RGB image, and the first layer uses a 3 × 3 volume Kernels, using 3x3 max pooling. The second layer uses a 3 × 3 convolution filter to filter the candidate boxes, and the third layer also uses a 3 × 3 convolution filter to further filter the candidate boxes. Through three convolutions and one pooling operation, the input 12×12×3 matrix is turned into a 1×1×32 vector, and finally a 1×1 convolution kernel is used to divide the output into face classification, bounding regression and key There are three parts of point positioning, and the structure diagram of P-Net is shown in Figure 2:

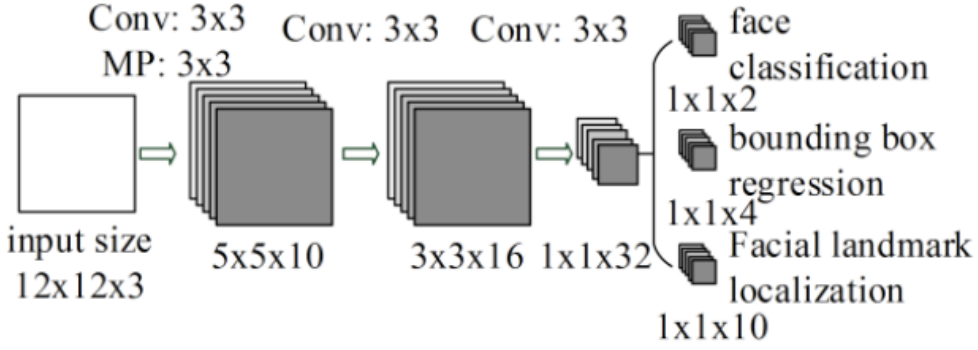


Figure 2. The architecture of P-Net,
where "MP" means max pooling and "Conv" means convolution.

The face classification in the first part is a binary classification problem of judging whether the candidate area contains a face, so the output is a 1×1×2 vector, that is, the candidate box contains a face or does not contain a face.

The second part of the bounding box regression mainly outputs the boundary coordinates of the candidate box, including the horizontal and vertical coordinates of the upper left corner of the boundary and the length and width of the bounding box. So the output is a 1×1×4 vector, which is the position of the border. Bounding box regression is a candidate box correction method widely used in object detection. The third part, Facial landmark localization, mainly outputs the coordinates of the facial key points, that is, the center position of the left eye, the center position of the right eye, the center position of the nose, the coordinates of the five points of the left and right mouth corners, each coordinates contain two parameters, horizontal and vertical, so the output is a 1×1×10 vector.

(2) R-Net is the second-layer convolutional network of MTCNN, the input is a 24×24×3 image, the first layer uses a 3×3 convolution kernel and uses 3×3 max pooling. The second layer filters the candidate frame through a 3×3 convolution kernel, still using 3×3 max pooling, the third layer uses a 2×2 convolution kernel to further filter the candidate frame, and finally connects a 128-dimensional full connection layer. The output is still divided into three parts: face classification, border regression and key point positioning. The structure diagram of R-Net is shown in Figure 3:

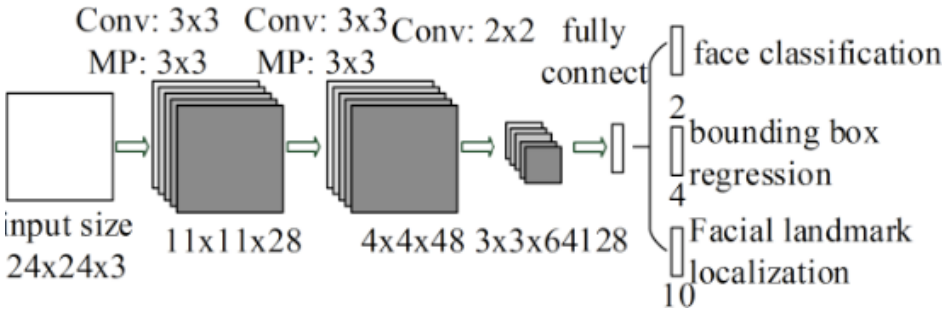


Figure 3. The architecture of R-Net

The working principle of R-Net is the same as that of P-Net. Based on P-Net, a more complex convolutional neural network is used, and the candidate frame is filtered by the non-maximum value suppression method of frame regression to reduce the number to be screened. the number of boxes, thereby reducing the amount of computation. Running

slower than P-Net due to more network layers.

(3) O-Net is the last network of MTCNN, similar to R-Net. The input image size is 48×48×3, the first layer uses a convolution kernel of size 3×3, and max pooling of 3×3. The second layer uses a 3×3 convolution kernel to filter the candidate frame, and still uses a 3×3 max pooling. The third layer uses a 3×3 convolution kernel to further filter the candidate frame, using a 2×2 Max pooling and then connecting a 256-dimensional fully connected layer. Finally, the results of face classification, border regression and key point positioning are obtained. The O-Net network structure is shown in Figure 4.

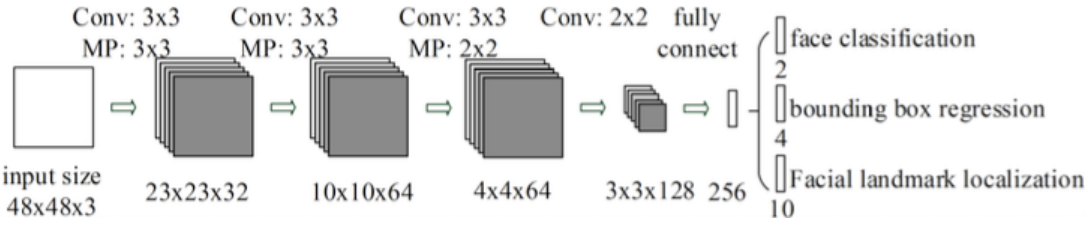


Figure 4. architecture of O-Net

Compared with R-Net, O-Net has one more layer of convolutional neural network. Due to the increased complexity of the network, the size of the input image and the parameters of each layer have increased, the accuracy has been greatly improved, and the ability to judge the input Further improvement, the network processing is more detailed, the output face detection frame is less, and the confidence level is also higher. The accuracy of key point detection is also higher.

MTCNN is a multi-task cascade network, which uses multi-task collaborative learning to train the network to achieve the purpose of face detection and key point location.

(1) *Face classificatio*n: it can be boiled down to binary classification with function of cross entropy. The calculation formular shows in formula 2.

$$L_i^{det} = -(y_i^{det} \log(p_i) + (1 - y_i^{det})(1 - \log(p_i))) \tag{2}$$

(2) *Border regression:* For each candidate box, predicts the offset from the real border. This is a regression problem, using Euclidean distance to calculate the loss.

$$\boldsymbol{L_i^{box} = ||\hat{y} - y_i^{box}||_2^2} \tag{3}$$

(3) *Facial landmark localization:* It is also a regression problem, uses Euclidean distance to calculate the loss.

$$L_i^{landmark} = ||\hat{\boldsymbol{y}}_i^{landmark} - \boldsymbol{y_i^{landmark}}||_{\boldsymbol{2}}^{\boldsymbol{2}} \tag{4}$$

(4) *Multi-source training:* For the calculation of the overall loss, each network of MTCNN has different functions and has different weights according to the distribution of samples during training. The overall loss function is as follows.

$$min \sum_{i=1}^{N} \sum_{j \in \{det,box,landmark\}} \alpha_j \beta_i^j L_i^j \tag{5}$$

To improve the detection accuracy, in each mini-batch sample, the forward propagation loss of all samples is sorted, the top 70% are selected as valid samples, and the gradient of back propagation is calculated. Then the simple samples are ignored, and the selection process of the samples is optimized to obtain better training results.

Chapter 3: Design and Implementation

After reviewing products related to mask detection, the authors propose a two-stage mask detector and develop an operator interface for it, a dataset for training and testing performance. In this chapter, the design and implementation of the system, face detector and the mask classifier will be introduced respectively. Finally, the datasets used to train the neural network are introduced.

3.1 System Design and Implement

To design this face mask identification system, the author divided it into 4 main modules, including Data Collection module, Data Process module, Model Process module and Storage module. The system design diagram is shown in Figure 5. There are several different functions in these modules.

First, the Data Collection module is a module for interacting with the user and collecting data. It has Image Collection function, Video Collection function and Real-time Collection function. These functions can read pictures or videos entered by the user or real-time camera.

The second part is the Data Process module. This module has two functions: image preprocess and dataset loading. Image preprocess is used to process the data passed in by the Data Collection module. It can resize the incoming image and convert it to RGB format. If the input is a video, each frame of the video is processed the same as image. The Dataset load module is used to process datasets. It can read images from the dataset and perform different preprocessing for training different networks. These processing mainly include IoU-based sample classification, image resize and processing datasets into csv files.

The third part is the Model Process module, which is also the core of the department. Two modules are included here: train module and predict module. In the train module, the PNet, RNet and ONet of MTCNN are trained sequentially by reading and preprocessing human face samples. The CNN mask classifier is then trained using mask-wearing faces and normal faces. The most suitable model is selected after convergence. for predict module. The Predict module will perform face detection on the preprocessed image, and then transmit it to the mask classifier after detecting the face. The mask classifier will classify faces into 'mask' and 'no mask'.

The last part is the Storage module. This part is used to store logs, models and datasets. The

Log management module can save logs of model training and user testing. These logs can be used to evaluate models and count user usage. The Model store module is used to store the trained Face Detector model and Mask classification model. The Dataset store module is used to store the dataset used for training.

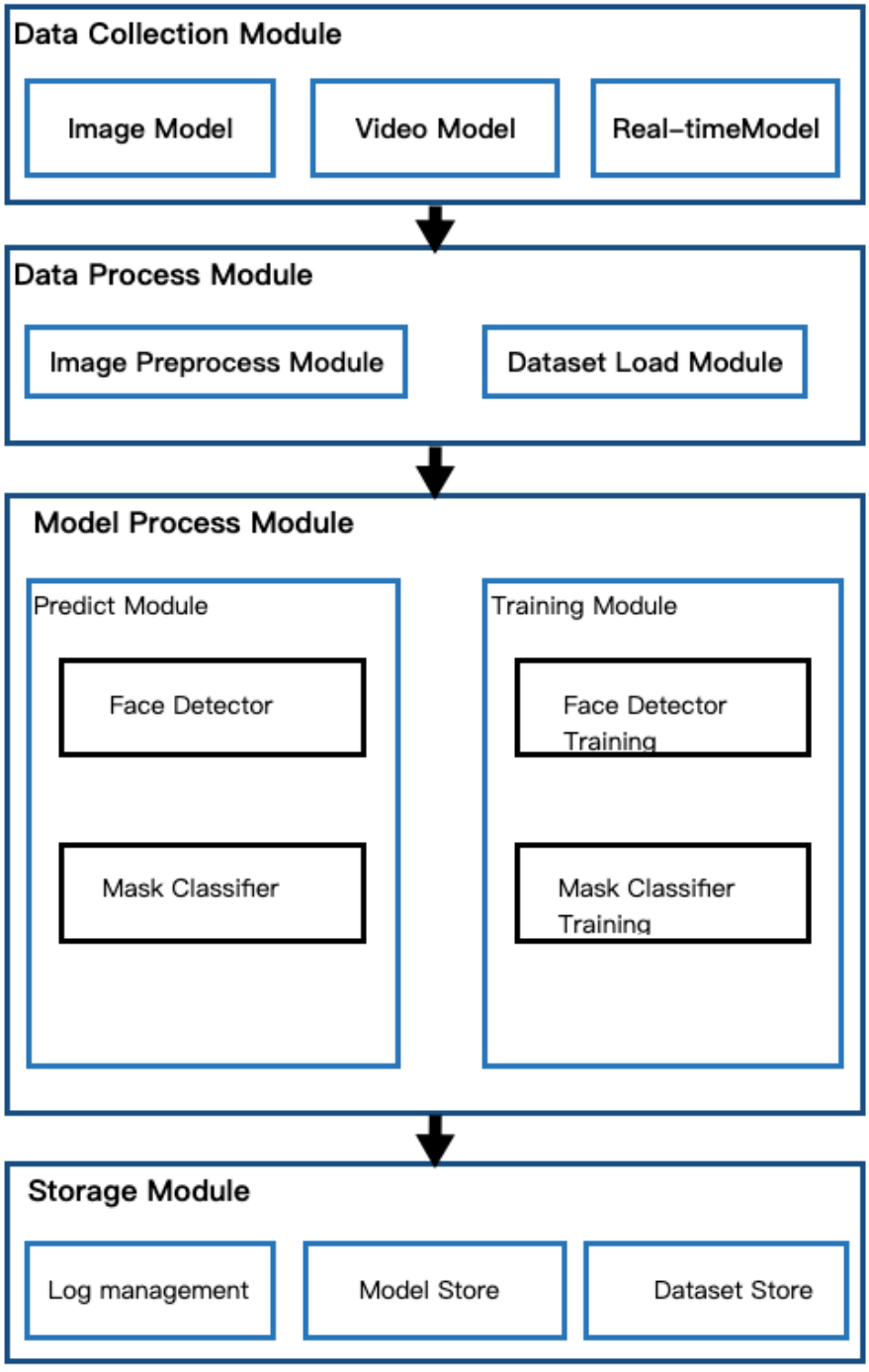


Figure 5. Design of System

The full process of this mask detection system is shown in the Figure 8. It provides users with 3 choices to upload video or image, or real-time camera for detection. If the user chooses image detection, the image will be passed to the FaceDetector to extract the faces, and if it is not extracted, the image will be displayed directly. If faces are extracted, each extracted face will be input into MaskDetector. If wearing a mask will be marked as "mask", if not wearing a mask will be marked as "no mask". Then put it in the input image and display it. If the user input is a video, the system will process each frame the same as the picture and display it after processing. For the convenience of the user, I developed a simple software that runs on the terminal to allow the user to select the detection mode. Figure 6,7 are screenshot of the software.

```
(pytorch) yorki@MacBook-Pro-Y FaceMaskDetection % python main.py
________Face Mask Detection__________
|       please choose model           |
|        You can choose:              |
|         1.Image Detection           |
|         2.video Detection           |
|         3.Real-time Detection       |
|         4.exit                      |
--------------------------------------
|Select your mode:                    |
```

Figure 6. Snapshot of detecting system

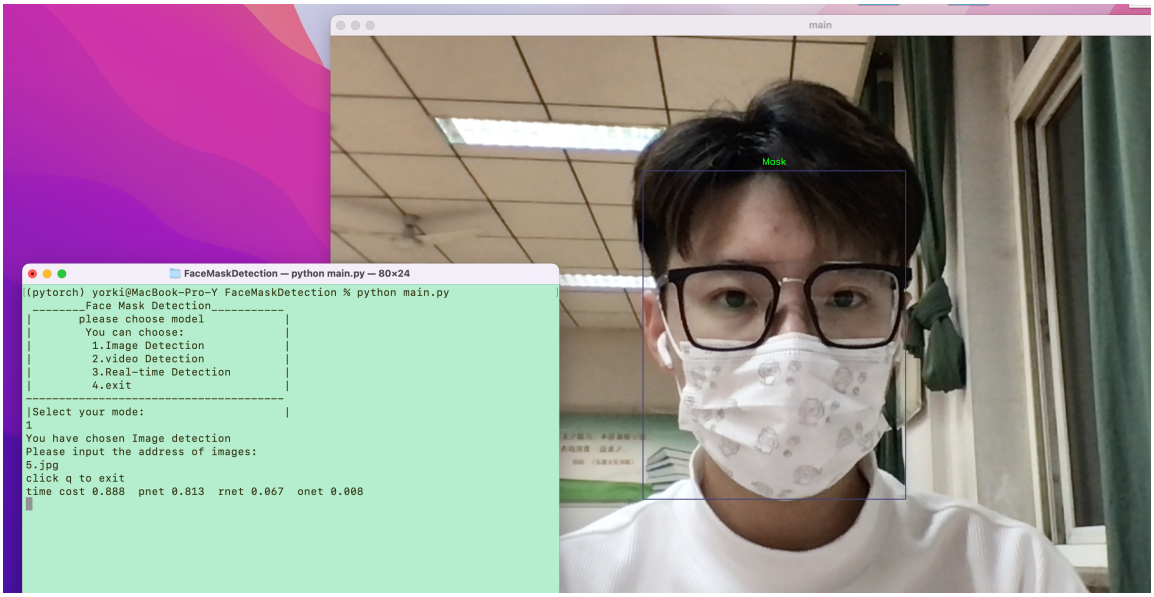


Figure 7. Snapshot of detecting system: image test mode

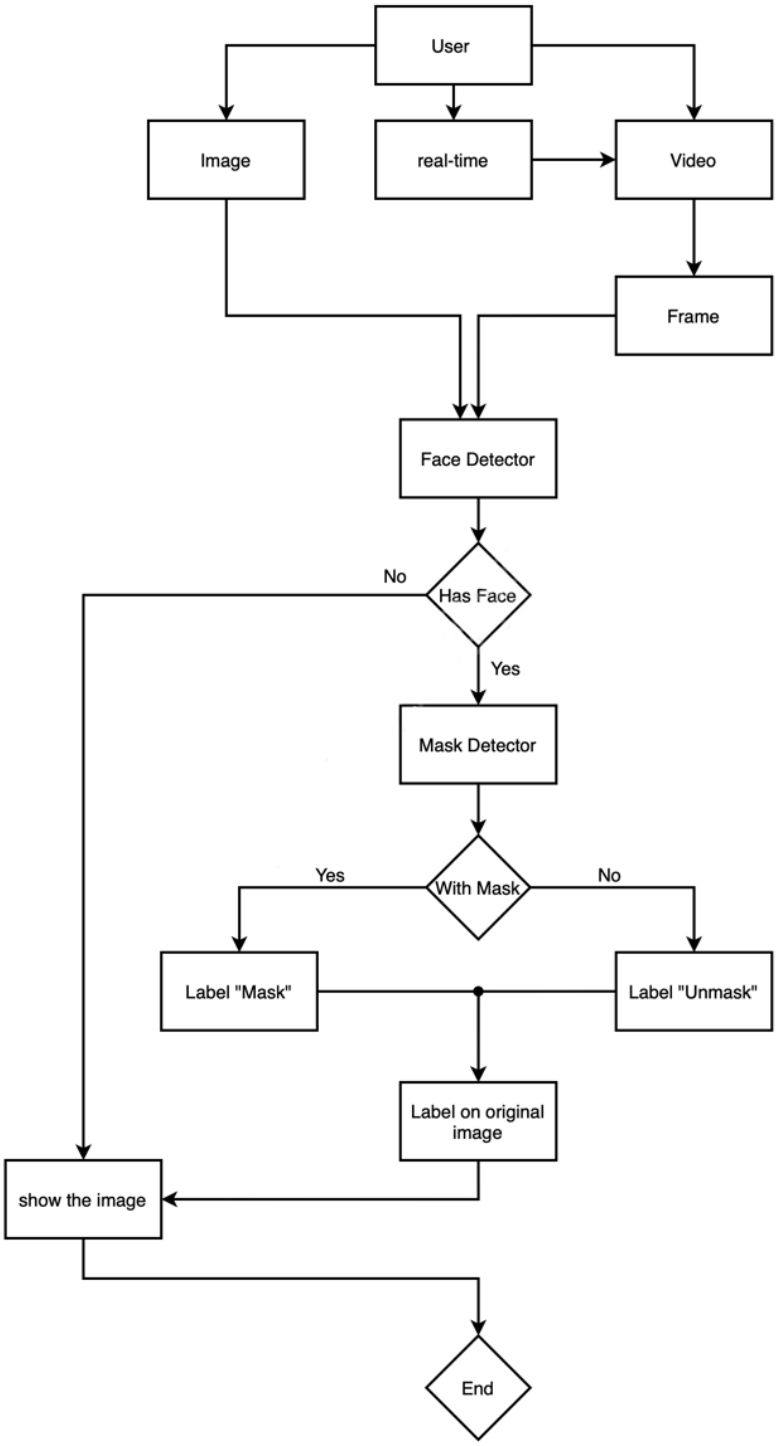


Figure 8. Workflow Facial Mask Wearing Identification System

3.2 Workflow of Detector

The overall flow of the face mask detection is shown in Figure 9. The image input will be

resized into 100*100 for later detection. The first stage is a face detector based on MTCNN. The MTCNN will process the image and locate the faces in image. Every face in the image will be extracted and delivered to the NN classifier. The classier takes the face and predicts whether there is a mask on it. And a bounding box with label ‘mask’ or ‘unmask’ will be showed in the face. Thanks to the lightweight of the model, it can process image, video and even real-time camera.

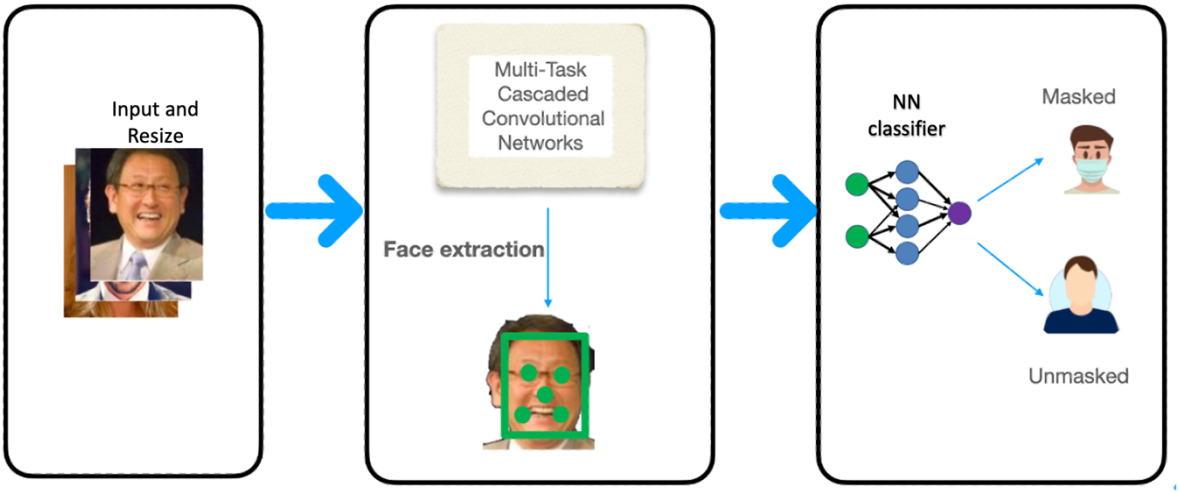


Figure 9. Workflow of DCNN

3.3 Face Detector

3.3.1 MTCNN

As mentioned in [10], MTCNN is a very powerful face detection network compared with YOLO v3. Figure 10 shows the simplified workflow of MTCNN face detection and alignment. Due to his unique three-stage structure and NMS mechanism, it can accurately locate the face. And it can accurately locate multiple human faces in a single image. Besides, it can overcome face missed detection, false detection and inaccurate key point positioning caused by low pixel and uneven illumination conditions. This satisfies our meeting for the detection of more than one people appearing in the collection camera at the same time in practical use.

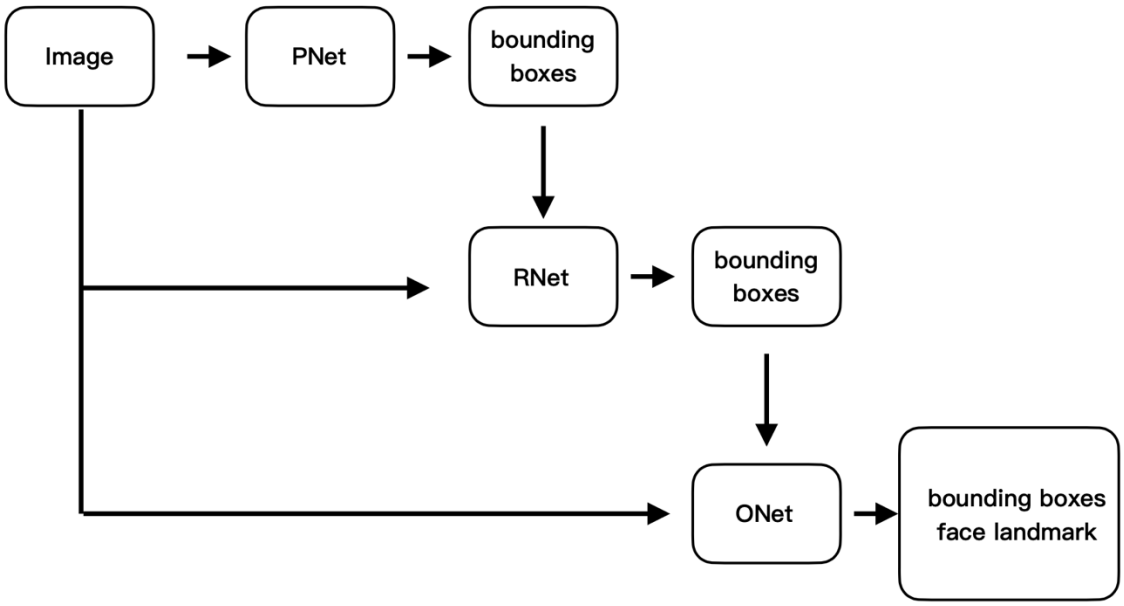


Figure 10. Workflow of MTCNN detection

3.3.2 Data Preparation

In this paper, four types of sample data are provided for the training of the improved MTCNN mask face detection network model, and different types of data are used to complete different model tasks. The four types of samples are: positive samples, negative samples, partial face samples and face key point samples. The three types of tasks are: face classification task, face region box regression task and face key point location task. The four types of sample training tasks and their definition standards are shown in Table 1. Besides, the data input size of P-Net is 12*12*3, that of R-Net is 24*24*3 and that of O-Net is 48*48*3. So, I generate the data from the dataset to meet the needs of different tasks and networks. Following [9], the author uses IoU (Intersection over Unio) to generate the data as well. The calculation of IoU is showed in Equation x.

$$IoU = \frac{Area\ of\ Intersection\ of\ two\ boxes}{Area\ of\ Union\ of\ two\ Boxes} \tag{6}$$

To make training pictures, the steps are as follows：

1. Read each image in the dataset and find out the location of its face based on the label
2. Calculate the center position of the face, randomly offset up, down, left and right, and form a square offset frame, and calculate its offset value.

3. Cut the image according to the newly formed box and scale it to the target size: 12, 24, 48
4. Calculate the IoU of the offset frame and the label face frame, and classify the clipped samples into positive samples, negative samples, and partial samples according to the value of IoU.
5. Save the information of images into csv for further training.

Table 1. Types of data

Type	Definition	Task
Positive	$IoU \geq 065$	Face classification task, bounding box regression
Negative	$\text{IoU} < 0.3$	Face classification task
Part faces	$0.45 \leq \text{IoU} \leq 0.65$	bounding box regression
Landmark faces		facial landmark localization

3.3.3 Model Training

The focus of each layer of network training tasks is different, so the corresponding weight coefficients are assigned to the loss functions of different tasks of the network at each stage.

Before imputing into network, the image will be prepossessed by randomly cropping and horizontal flipping to augment data. During training, I refer to the open-source training parameters in the network to reduce the time to adjust parameters and save computing resources. I set the batch size to 64 and set the learning rate to 0.01 for precise training. Besides, I set the train epoch to 10.

Table2: Parameters of training MTCNN

	PNet	RNet	ONet
End_Epoch	11	9	10
Learning rate	0.01	0.05	0.005
Batch Size	64	640	640
Lr_decay	9	8	8

3.4 Face Mask Classifier

3.4.1 Architecture of CNN

By using the improved MTCNN mask face detection algorithm in the previous chapter, the exact position of the rectangular frame of the face can be detected, and then the image of the face area is cropped based on the coordinate information of the face frame, and the constructed mask image classification network is used to identify Whether the face is wearing a mask, so as to meet the ultimate demand for mask face detection. In this part, the author designs a simplified neural network which contains 3 convolution layers and 2 linear layers, and chooses ReLU as the activation function, Maxpool for pooling layers. The architecture is showed in Figure 11, and parameters are showed in Table 3.

Table 3 Parameters of NN Classifier

Type	Input channels	Output channels	kernel	padding
Conv 1	3	32	3x3	1x1
MaxPool			2x2	
Conv 2	64	128	3x3	1x1
MaxPool	–	–	2x2	–
Conv 3	64	128	3x3	1x1
MaxPool	–	–	2x2	–
Linear	2048	1024	–	–
Linear	1024	2	–	–

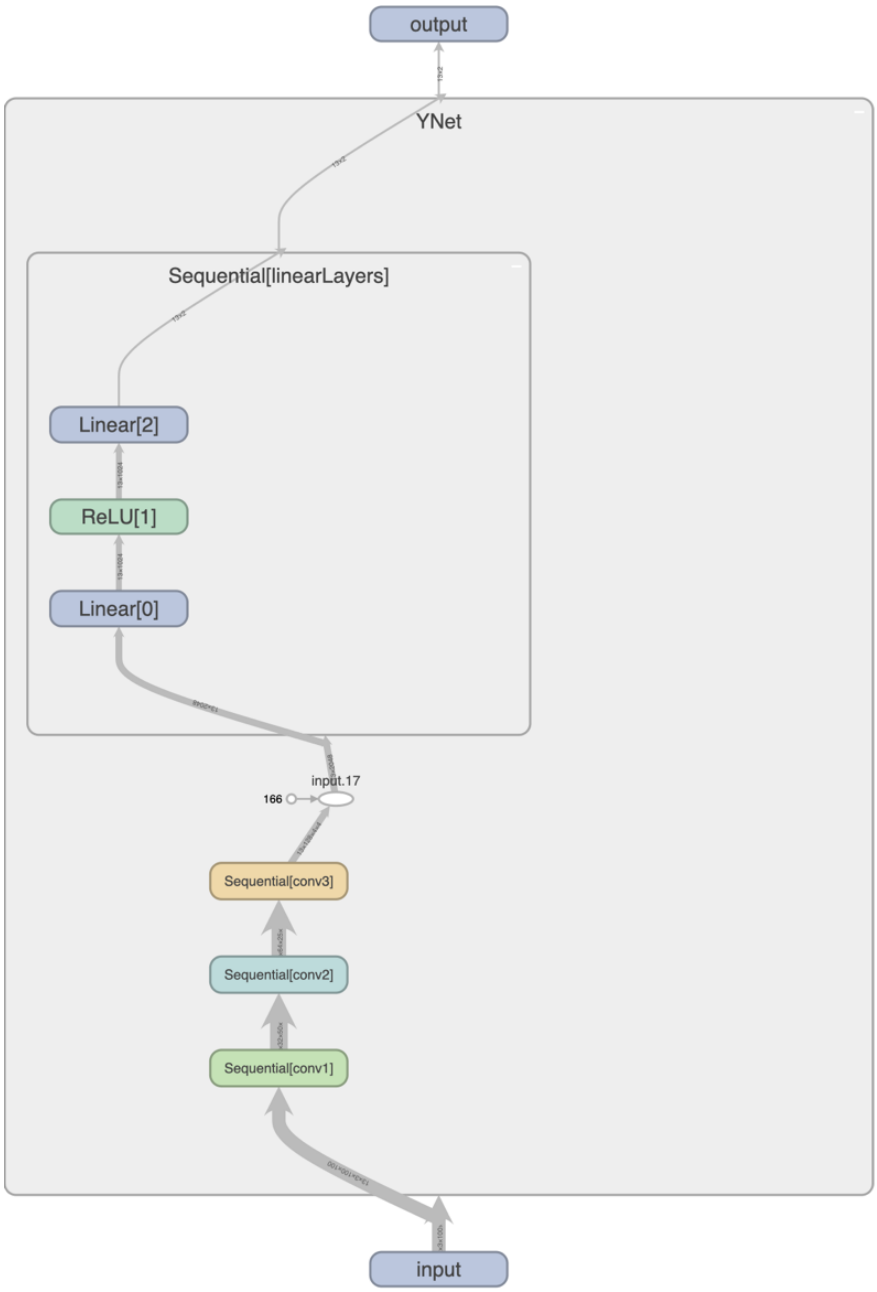


Figure 11. Structure of CNN Classifier

3.4.2 Data Preparation

As introduced in 3.1, the dataset used to train the mask classifier has 15,000 images of wearing masks but 90,000 images of normal faces. Due to the different numbers of positive and negative samples, it may be possible if a random selection is made will result in the disproportion of training step. So, when we split the dataset into training and validation sets, we need to keep

the same proportions of the images in train/validation as the whole dataset. To optimize it, I assign more weight to classes with a small number of samples and less weight to a large number of samples by Equation 7. In this way, the network will be penalized more if it makes mistakes predicting the label of small classes as well as makes the network training agnostic to the proportion of classes.

$$class\_weight = 1 - \frac{Class\_Cardinality}{\sum All\_Classes\_Cardinalities} \tag{7}$$

3.4.3 Model Training

In this paper, the mask classification model training method uses the weighted cross entropy loss function to propagate and adjust the model parameters, uses the Adam algorithm to optimize the model training results, and uses the xavier_uniform to initialize the weights. At the same time, the cross-validation method is used. 70% of the training samples are used for training the model, and 30% are used for model testing. The training parameters are showed in Table 4. To reduce the time-consuming of loading images, I use a multi-threaded method to speed up data loading and perform data enhancements such as flipping and rotating the training sample images, which effectively improves the accuracy and robustness of the network model. During training, I used Tensorboard to record the relevant data of the training. The training results are shown in Figure 12. The network model converges very quickly in the early stage, the accuracy rate is maintained at the level of 99.5%, and the total loss is gradually reduced and maintained at a normal level.

Table 4. Parameters of training CNN

	Parameters
Epoch	10
Learning rate	0.0001
Batch Size	32

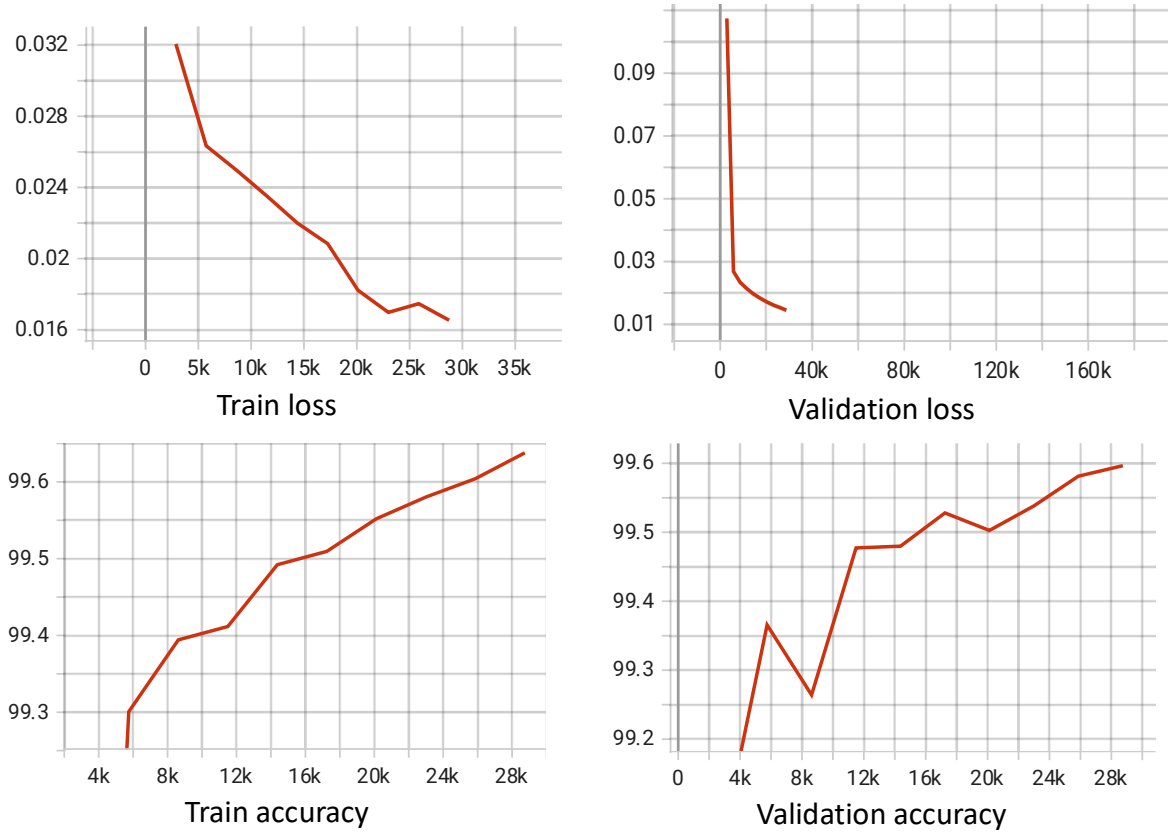


Figure 12. Loss and accuracy during training

3.5 Dataset Collection

In this paper, the author utilizes mainly two datasets for training and testing. For the face detector, the author uses WIDER FACE and Celeb A to train. For mask classifier, the author uses Real-World-Masked-Face-Dataset (RWMF) and a modified LFW dataset. Besides, the author use part of MAFA dataset and LFW as the test set for mask classifier. The overall status of dataset is showed in Table 5.

Table 5: Dataset

Dataset	Image content	Usage	Number of images
WIDER FACE	Human faces	Face detection	32,203
Celeb A	Human faces	Face detection	202,599
RWMF	Masked & unmasked	Mask classifier	5,000 masked 90,000 unmasked
Masked LFW	Masked faces	Mask classifier	10,000

MAFA	Masked faces	Test	1000

WIDER FACE [11]. This is one of the most widely used benchmark training datasets in the field of face detection and was provided by a research team at the Chinese University of Hong Kong in 2016. It contains 32,203 images and 393,703 labeled faces. The author selected 61 categories of events from WIDER and randomly selected 40%, 10% and 50% for each as a training, verification, and test set. Each subset contains three levels of detection difficulty: Easy, Medium, and Hard. These faces vary widely in scale, posture, illumination, expression and occlusion. The images selected by WIDER FACE are mainly sourced from public data sets. Figure 13 is samples of WIDER FACE dataset.

Figure 13. WIDER FACE dataset samples

Celeb A [12]. This is a face data set organized by the Chinese University of Hong Kong, which contains a total of more than 202,599 face images. Each image is named by the name of the person and contains the coordinate information of the face region and the coordinate information of the five key features of the face. This paper uses this dataset to train the face key point regression task. Figure 14 shows samples of Celeb A dataset.

Figure 14. samples of Celeb A

RMFD [13]. This is a mask data set compiled and published by Wuhan University. After sorting, cleaning and labeling, 5,000 masked images and 90,000 normal face images from 525 people have been collected from samples collected from the Internet. Figure 15 shows some samples of it.

Figure 15. RWMF dataset samples

Modified LFW. LFW (Labeled Faces in the Wild) [14] is a dataset compiled by the University of Massachusetts Amherst, which is mainly used to study unconstrained face recognition. The

LFW database, which mainly collects images from the Internet rather than the lab, contains more than 13,000 images of human faces. Considering that the number of photos of people wearing masks in the RMFD is too small compared to those who do not wear masks, the author creates some samples of people wearing masks based on LFW. MaskTheFace is an open-source software from Github, which can add masks to facial images automatically. Using it, the author adds surgical mask, cloth mask and N95 mask to faces, and randomly chooses 10,000 of them to extend the RMFD. Figure 16 shows samples of modified LFW.

Figure 16. Masked LFW dataset samples. In this dataset, the mask has types of surgical mask(left), cloth mask(mid), KN95 mask(right). This enhances the robust of mask classification.

MAFA [15] This is a masked face detection benchmark dataset, of which images are collected from Internet images. MAFA contains 30,811 images and 35,806 masked faces. Faces in the dataset have various orientations and occlusion degrees, while at least one part of each face is occluded by mask. In this paper, the author uses part of it as the test set.

3.6 Experimental Setup

During the training, the hardware and software experimental environment is showed in Table 6

Table 6: Setup of Experimental

Equipment	Model
CPU	Intel Core i7

Memory	16G
Operating system	Linux
Graphics	NVIDIA GeForce RTX 2080 Ti
Deep learning framework	Pytorch

Chapter 4: Results and Discussion

4.1 Accuracy Measures

This paper uses Accuracy, precision, recall to evaluate the detection effect and performance of the model. The following introduce it in detail.

- Accuracy (A): the number of correctly classified instances of mask and no mask over the total number of instances using the following Equation 8.

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

- Precision: the number of correctly detected class members by the classifier over the total number of correctly and incorrectly detected members using Equation 9.

$$P = \frac{TP}{TP + FP} \tag{9}$$

- Recall: the proportion of the positive class predicted by the classifier and the true label is positive class to all the true label positive class.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

In these equations, the parameters' meaning is:

TP (True Positives): The positive samples correctly classified by the detection model.

TN (True Negatives): The negative samples correctly classified by the detection model.

FP (False Positives): Positive samples that are misclassified by the detection model.

FN (False Negatives): Negative samples that are misclassified by the detection model.

4.2 Result and Analysis

4.2.1 Face detector analysis

In this paper, the author uses the WIDER FACE validation set to test the face detector performance. Compare with the result of [9], the model trained performs better in easy, medium, hard situation. This enables the face mask detection system can perform well in idea environment as well as bad situation such as covered, weak light etc. This enhancement will allow model to have a wider range of scenarios when testing masks. It also allows us to detect

multiple faces simultaneously under harsh conditions.

Table 7 Compare trained model and original model

Style	Easy	Medium	Hard
MTCNN-original	65.3%	65.1%	40.3%
MTCNN-trained	71.4%	70.4%	43.2%

Figure 17. Some results of MTCNN detector

4.2.2 Face mask classifier analysis

This paper uses 2000 images for model testing, including the MAFA dataset randomly selecting 1000 images with masks, and the LFW dataset selecting 1000 face images without masks. In order to fully evaluate the training model effect, this paper uses the introduction of the previous

chapter. The classification evaluation index of the output experimental results.

Table 8: Result of test.

Name	Meaning	Value
TP	Predict value 1, real value 1	946
TN	Predict value 0, real value 0	917
FP	Predict value 1, real value 0	83
FN	Predict value 0, real value 1	54
Accuracy	-	93.15%
Precision	-	91.93%
Recall	-	94.6%

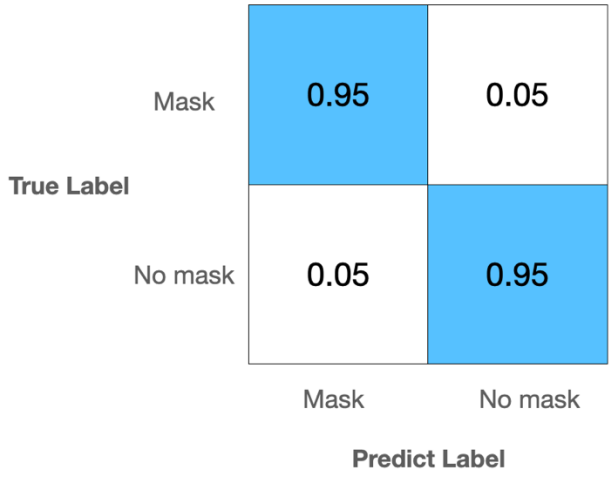


Figure 18. Confusion matrix of model. Accuracy: 0.93

It can be seen from the evaluation indicators of image two-classification that the accuracy of the trained mask image two-classification model can reach 94%, and the classification effect meets the needs of mask wearing image classification under daily monitoring conditions. An example of a test image using the trained network model is shown in figure. And compared with the Vihn's model based on YOLOv3, the performance of proposed model is better than that in accuracy, precision and recall. The detection of the system in poor light and low resolution still needs to be improved. Especially because the occlusion of face detection after wearing a mask will cause missed detection of face detection.

Table 9: Compared with Vihn's model [3]

	Vinh's model	Proposed
Accuracy	90.1	93.1%
Precision	83%	91.9%
Recall	90.1%	94.6%

Figure 19. Some results of mask detection

Chapter 5: Conclusion and Further Work

5.1 Conclusion

This paper analyzes and studies the detection of mask wearing in the process of epidemic prevention in daily life under the background of the new coronavirus. In order to effectively avoid the risk of cross-infection in manual detection, the author proposes to utilize the computer vision combined with deep learning algorithms to construct a face mask wearing detection system. Based on the existing model and datasets, the author modifies them to train a lightweight model that meets the performance requirements of practical use. The main content and research results of this paper are summarized as follows.

1. This paper analyzes the era background and research significance of face mask wearing detection from the perspective of daily epidemic prevention of Covid-19; and analyzes the existing detection methods. Depending on the great achievements of deep learning in face recognition, the author proposes the idea to develop a two-stage face mask wearing detection using deep learning algorithm. This project is made extremely modifiable by separately training the two-stage model. Either the face detection and extraction model from the first stage can be used in other research related to face detection. The classifier of the second part can also be modified to a face mask classification model trained by other networks.

2. Construct a face mask dataset of nearly 100,000 images. This dataset collects 5,000 face pictures with masks and 90,000 normal face pictures in the Real-World Masked Face Dataset (RMFD), an open-source face annotation dataset. But the photos in the RMFD are not balanced enough, and there are fewer pictures of wearing masks. The author utilizes the Labeled Faces in the Wild (LFW) dataset and adds simulated masks to these images using an open-source software on Github called MaskTheFace. These simulated masks include the vast majority of mask types on the market, enhancing the richness of the dataset and the ratio of positive and negative samples.

3. A human face detection model. This paper introduces the MTCNN widely used in the field of face detection and introduces its core process. Using the WIDER FACE dataset and open-source training parameters, the author trains a model that can accurately extract facial features under different lighting, image quality and other conditions. The model is trained

separately, so it can be used independently by other researchers in other face-related experiments.

4. A lightweight face mask classifier model. The mask image classification task is performed on the face images output in the first stage of mask wearing detection. This paper constructs a neural network with three convolutional layers and two linear layers, uses the modified RMFD dataset and conducts network training. The experimental results show that the network model has high accuracy and stability, and has achieved a good mask image classification effect, which can meet the requirements of practical situations.

5.2 Further Work

In the mask image classification stage, this paper only proposes a simple two-classification process, that is, the state of wearing a mask and not wearing a mask. In daily life, there are often scenes where masks are not properly worn. The network model in this paper will predict this type as wearing a mask, which is a wrong prediction result in the process of epidemic prevention. In order to keep the model lightweight, the mask classification neural network proposed in this paper has few parameters, sacrificing part of accuracy. Parts of the face can be misjudged or missed. In the future research process, it is hoped that the classification task of not wearing a mask can be solved, and the precise positioning of the classification task can be improved. At the same time, when other occludes are used for face occlusion, the detection results of wearing masks will also result.

References

[1] S. E. Snyder and G. Husari. (2021). Thor: A Deep Learning Approach for Face Mask Detection to Prevent the COVID-19 Pandemic. SoutheastCon. 2021, pp. 1-8.

[2] S. Susanto, F. A. Putra, R. Analia and I. K. L. N. Suciningtyas. (2020). The Face Mask Detection For Preventing the Spread of COVID-19 at Politeknik Negeri Batam. 2020 3rd International Conference on Applied Engineering (ICAE). pp. 1-5.

[3] T. Q. Vinh and N. T. N. Anh. (2020). Real-Time Face Mask Detector Using YOLOv3 Algorithm and Haar Cascade Classifier," 2020 International Conference on Advanced Computing and Applications (ACOMP). pp. 146-149.

[4] aqeelanwar, “Masktheface.” https://github.com/aqeelanwar/ MaskTheFace, 2020.

[5]World Health Organization, “novel-coronavirus-2019” https://www.who.int/emergencies/diseases/novel-coronavirus-2019.

[6] S. Ren, K. He, R. Girshick and J. Sun. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. 39, no. 6, pp. 1137-1149.

[7] R. Girshick, J. Donahue, T. Darrell and J. Malik. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 580-587.

[8] J. Redmon and A. Farhadi. (2018). Yolov3: An incremental improvement. CoRR. vol. abs/1804.02767.

[9] Zhang, K., Zhang, Z., Li, Z. and Qiao, Y.. (2016). Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks. IEEE Signal Processing Letters, 23(10), pp.1499-1503.

[10] N. Zhang, J. Luo and W. Gao. (2020). Research on Face Detection Technology Based on MTCNN. 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA). pp. 154-158

[11] S. Yang, P. Luo, L. C. Change, and X. Tang. (2016). Wider face: A face detection benchmark. IEEE CVPR. pp. 5525– 5533

[12] [32] S. Yang, P. Luo, C. C. Loy, and X. Tang. (2015). From facial parts responses to face detection: A deep learning approach. IEEE ICCV. pp. 3676–3684, 2015.

[13] Wang, Zhongyuan & Wang, Guangcheng & Huang, Baojin & Xiong, Zhangyang & Hong, Qi & Wu, Hao & Yi, Peng & Jiang, Kui & Wang, Nanxi & Pei, Yingjiao & Chen, Heling & Yu, Miao & Huang, Zhibing & Liang, Jinbi. (2020). Masked Face Recognition Dataset and Application.

[14] G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller. (2012). Learning to align from scratch. NIPS

[15] S. Ge, J. Li, Q. Ye and Z. Luo. (2017). Detecting Masked Faces in the Wild with LLE-CNNs. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 426-434

Acknowledgement

I've read a lot of acknowledgements and pictured myself writing them a lot, but I never anticipated how relaxed I'd feel doing them.

The primary concern is a project. Thank you to my supervisor for responding so quickly after I wrote the email, and the criterion for choosing the topic is: the first responder counts. I'd want to express my gratitude to the Supervisor for his assistance with the project. Despite the fact that I was late and slow on the assignment, he patiently provided me with a lot of helpful advice to help me realize the model and write the report. In addition, I was still studying for the TOEFL, GRE, and application materials while working on the project. My supervisor was supportive of my efforts and encouraged me to keep going.

At the same time, I'd like to express my gratitude to the other members of the group. When we talked about the project, I picked up a lot of new information. Finally, I'd want to express my gratitude to my roommates and some more experienced friends for their unselfish responses to some simple and foolish queries, which not only inspired me to keep going, but also gave me hope that I might still make programming as my career.

Appendix

- Specification

  Part 1

北京邮电大学 本科毕业设计(论文)任务书

Project Specification Form Part 1 – Supervisor

论文题目 **Project Title**	Facial Mask Wearing Identification System Based on Deep Learning		
题目分类 **Scope**	Data Science and Artificial Intelligence	Research	Software
主要内容 **Project description**	Wearing facial masks is an effective way to reduce Covid-19 spread. It takes time and effort to check and remind everyone who enters a building to wear a facial mask. The goal of this project is to create a deep learning model that can detect whether a person is wearing a facial mask based on a photo taken by a camera. The designed model would be trained on collected datasets containing human face images with and without facial masks using the supervised learning method. To optimize the model, the stochastic descend algorithm or its variants will be used. Pytorch is used in the training and testing environment. Finally, the effectiveness of the designed model will be assessed using 10-fold cross validation.		
关键词 **Keywords**	Deep Learning, Convolutional Neural Network, Image Classification		
主要任务 **Main tasks**	**1** Collecting images containing human faces with and without facial masks.		
	2 Design the architecture of a convolutional neural network for detecting facial mask		
	3 Write the code for training and testing the designed neural network on the collected images.		
	4 Write a report that describes the design and implementation of the neural network and experiment results.		
主要成果 **Measurable outcomes**	**1** A dataset containing at least 20,000 human face images with or without facial masks.		
	2 A trained neural network model that can detect offline if a given photo contains a face wearing a facial mask.		
	3 A software based on the model to detect online if a person is wearing a facial mask.		

Part 2

北京邮电大学 本科毕业设计（论文）任务书
Project Specification Form
Part 2 - Student

学院 School	International School	**专业 Programme**	**Telecommunications Engineering with Management**		
姓 Family name	Yang	**名 First Name**	Yongqi		
BUPT 学号 BUPT number	2018212807	**QM 学号 QM number**	190015176	**班级 Class**	2018215108
论文题目 Project Title	Facial Mask Wearing Identification System Based on Deep Learning				

论文概述 **Project outline** **Write about 500-800 words** **Please refer to Project Student Handbook section 3.2**	**An initial analysis of user requirements.** COVID-19 coronavirus epidemic has spread all over the world. Wearing face mask is an effective way to protect people. Arranged a staff to remind people to wear masks at the door is costly and inefficient. To overcome this situation, I will implement a software with a robust facial mask detector using deep learning to cut the cost. The main function of the software is to monitor whether a person is wearing a mask in real time and mark it. It can also mark people who are not wearing masks correctly. If it is found that someone is not wearing a mask or is not wearing a mask correctly, the software will alert the user by voice. Its accuracy will be above 92% and the recognition can be completed within two seconds. Besides it can identify multiple users at the same time to improve efficiency. It can record how many people masked, unmasked, and masked incorrectly, and display this data on the screen, which is beneficial to statistics of the population. **The algorithms, methodologies, and other techniques to be employed** Basically, I will use deep learning to train the facial mask detector and use Pytorch as training and testing environment. Firstly, I will make a dataset consist of images of people with mask and without mask. It will have above 80,000 images for training and testing. I will download the established datasets first from the Internet, including ImageNet, kaggle, Real-World-Masked-Face-Dataset, etc. To enlarge the dataset, I will download face photos from the Internet through a web crawler and add analogy masks to them. Because there are already many trained models with similar functions, I will use them to label the data set. Then I will clean the data and use LabelImg to relabel the incorrectly labelled photos. Secondly, I will use image pre-process the images. I will do histogram equalization to the picture and then apply rotation and scaling to the images. Thirdly, when training the model, I will use COCO dataset to construct a pre-trained model that detects human subject. Then I will apply we utilize the Multi-Task cascaded Convolutional Neural Network classification and apply it to dataset so that It can detect face accurately. At last, I will use MobileNetV2 to construct a convolutional neural network model to classify instances into mask or unmasked. To evaluate the performance, I will measure models from accuracy, precision, recall, F1 score. Further, load the model and detract the face ROI is extracted. It will be a rectangular shape mounted automatically to cover the face, hair, and neck of the models. And then pre-process it as I have done during the training. Finally, the face mask detector is applied, and the images classified as with masked, unmasked and masked incorrectly. I will implement a web app to interact with users. I will use Flask to build the back end of the app to call the model and record the number of users and accuracy. And I will use Vue.JS and Bootstrap to build the front-end interface.

	An initial specification of how users will interact with the system(implementation) I will develop a web app for users to upload video or picture and identify whether they are wearing a mask. Besides that, it can draw a line graph to show the daily flow of people and a pie chart that can show the proportion of the number of people wearing masks. **Experiments that should be done to prove the project hypotheses(research)** I will use 10-fold cross validation to verify the result and draw some plots to show my results such as loss, ROC curve, accuracy etc. **Programming language, database software package to be used** Python, Flask, HTML, CSS, JavaScript, Vue.JS, Bootstrap, Mysql, OpenCV, Pytorch, **A list of background material consulted including webpages** 1. I. B. Venkateswarlu, J. Kakarla and S. Prakash, "Face mask detection using MobileNet and Global Pooling Block," 2020 IEEE 4th Conference on Information & Communication Technology (CICT), 2020, pp. 1-5 2. S. E. Snyder and G. Husari, "Thor: A Deep Learning Approach for Face Mask Detection to Prevent the COVID-19 Pandemic," SoutheastCon 2021, 2021, pp. 1-8 3. S. I. Ali, S. S. Ebrahimi, M. Khurram and S. I. Qadri, "Real-Time Face Mask Detection in Deep Learning using Convolution Neural Network," 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), 2021, pp. 639-642 4. S. A. Sanjaya and S. Adi Rakhmawan, "Face Mask Detection Using MobileNetV2 in The Era of COVID-19 Pandemic," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-5 5. K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint face detection and alignment using multitask cascaded convolutional networks,” IEEE Signal Processing Letters, vol. 23, no. 10, pp. 1499–1503, 2016.
道德规范 **Ethics**	Please confirm that you have discussed ethical issues with your Supervisor using the ethics checklist (Project Handbook Appendix 1). [YES/NO]

	Summary of ethical issues: (put N/A if not applicable)
中期目标 **Mid-term target.** **It must be tangible outcomes, E.g. software, hardware or simulation.** **It will be assessed at the mid-term oral.**	1. Construct a dataset which has over 80,000 labelled images. The number of photos of people wearing a mask and not wearing a mask is close to 1:1. 2. Build a web app that can capture video in real time. Implement the people counting function on the front-end page, draw a line graph to show the daily flow of people and a pie chart that can show the proportion of the number of people wearing masks. Provide an interface for the mask detector in the backend. 3. Pre-train a model that can recognize faces. Its recognition accuracy can reach more than 98%. Processing time is within 0.5s

Work Plan (Gantt Chart)

Fill in the sub-tasks and insert a letter X in the cells to show the extent of each task

	Nov 1-15	Nov 16-30	Dec 1-15	Dec 16-31	Jan 1-15	Jan 16-31	Feb 1-15	Feb 16-28	Mar 1-15	Mar 16-31	Apr 1-15	Apr 16-30
Task 1 Collecting images containing human faces with and without facial masks.												
Download the images containing human faces	X											
Classify them into with and without mask		X										
Label them for further using			X	X								
Task 2 Design the architecture of a convolutional neural network for detecting facial mask												
Study deep learning, OpenCV and Pytorch basic.	X	X	X	X								
Read relative paper and reiteration them			X	X	X	X						
Choose proper method and design						X	X					
							X	X				
Task 3 Write a report that describes the design and implementation of the neural network and experiment results.												
Write code for pre-processing the images							X	X				
Write code for training and testing.								X	X	X		
Plot the accuracy and loss									X			
Task 4 Write a report that describes the design and implementation of the neural network and experiment results.												
Write the outline of the report								X				
Implement the python GUI for interacting with user					X	X						
Complete the report with the results of model									X	X	X	X

- Early-term Progress Report

北京邮电大学 本科毕业设计（论文）初期进度报告

Project Early-term Progress Report

学院 School	International School	**专业 Programme**	**Telecommunications Engineering with Management**		
姓 Family name	Yang	**名 First Name**	Yongqi		
BUPT 学号 BUPT number	20182121807	**QM 学号 QM number**	190015176	**班级 Class**	2018215108
论文题目 Project Title	Facial Mask Wearing Identification System Based on Deep Learning				

已完成工作 Finished work:

Summary of material was read or researched

1. Thor: A Deep Learning Approach for Face Mask Detection to Prevent the COVID-19 Pandemic

   In this paper, it implemented a detector that extracts powerful features from low-quality images taken by a mobile robot to construct a classifier that detects unmasked personnel with high accuracy. The main factors that affect feature extraction from images are the height difference between the camera and the face, angle between the camera and the face, quality of light and distance to human subjects. Different with other approaches to detect masks, it does not function based on the assumption that people are facing the camera and are only a few inches away like most popular datasets. The detector works with footage (videos) with varying lighting quality as indoor spaces have different lighting intensities which affects the quality of captured images.

   The implementation of the detector is in order of data collection and preprocessing, human subject detection, face detection and extraction and face mask classification.

   In data collection and preprocessing part, to reduce the size of the data, its sampler selected only one frame from the 20 frames captured in each second and discarded the other 19 images captured during that second. This process reduced the size of our data to 5% and boosted the performance of the following detection modules by 95%.

   In human subject detection part, it utilizes RetinaNet to automatically extract human subjects and filter out irrelevant content from the dataset. And it can detect all 229 human subjects achieving a recall of 100%. RetinaNet is an architecture that integrates ResNet with Feature Pyramid Network (FPN). Moreover, it uses a Feature Pyramid Network backbone on top of a feedforward ResNet (particularity ResNet-50) architecture, for image recognition.

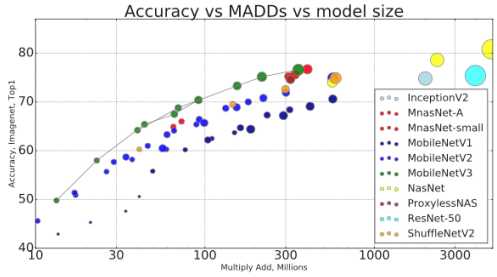
In face detection and extraction part, the researchers utilize the Multi-Task cascaded Convolutional Neural Network (MTCNN) classification to the dataset and label images into “face” or “no face”. These labels are assigned confidence scores by MTCNN. We explain in the next section how we further utilize these scores for mask detection. The MTCNN classifier achieved an accuracy of 94.4%.

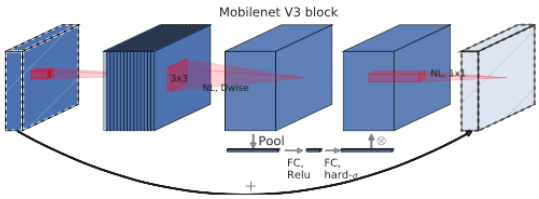
In face mask classification part, the researchers construct a convolutional neural network model to classify instances into mask or unmasked using MobileNetV2. It is trained by a public Custom Mask Community Dataset (CMCD) which has 690 masked facial images and 686 unmasked facial images. They used 80% of dataset for training and 20% for testing, and it achieved 99% accuracy.

To evaluate the performance of model, the researchers investigated its accuracy, precision, recall, and f1 score under different settings. They manually labeled 133 facial images as “masked” and 65 as “unmasked”. The detector agreed 161 times (or 81.31%) with our manual labeling. It correctly detected 132 masked faces out of 133 possible ones achieving a recall of 99.24%. Also, it classified 37 images (or 18.68%) differently than our labeled set such that it falsely classified 1 image of a masked face as “unmasked” achieving a false negative rate that is less than 1% (precisely 0.75%).

2. Searching for MobileNetV3

In this paper, it presents the 3rd generation of MobileNets based on a combination of complementary search techniques as well as a novel architecture design. It released 2 versions of MobileNet v3, MobileNetV3-Large and MobileNetV3-Small which are targeted for high and low resource use cases. In short, V3-large was more accurate and slower than V3-small. In the detection task, v3-Large achieves the same accuracy as V2, and the speed is increased by 25%; In the segmentation task, v3-large with LR-ASPP achieves the same accuracy as V2 with R-ASPP, and the speed is increased by 30% as the following figure shows.





For efficient mobile building blocks, MobileNetV3 combines the *depthwise separable convolutions* in MobileNetV1, *linear bottleneck and inverted residual structure in* MobileNetV2 and *lightweight attention modules based on squeeze and excitation into the bottleneck structure* in MnasNet. It used the hard sigmoid to improve the efficiency of computing by squeeze and excitation as well as the swish nonlinearity. The structure of it is showed above.

To improve the network, the paper proposes 2 main methods: Redesigning

expensive layers and Nonlinearities. For *redesigning expensive layers method*, the first is the modification of the most backward part, which is the modification of the prediction part. In mobilenet-V2's model architecture, the last part of the network is first mapped to high dimensions by 1x1 convolution, then features are collected by GAP, and finally divided into K classes by 1x1 convolution. So, the one that plays the role of feature extraction is the one that does 1x1 convolution at 7x7 resolution. According to the above idea, in MobileNetV3, features are firstly pooled, then extracted by 1x1 convolution for training the features of the final classifier, and finally classified into class K. With this improvement, 15% of the time (~10ms) was saved, and about 30 million multiplication and addition operations were reduced, but almost no accuracy was lost. Existing mobile terminal models tend to use 32 standard 3*3 convolution to build the initial filter bank. From a qualitative perspective, the role of this filter bank is to detect edges. Thanks to Hard Swish's design, it was possible to reduce the number of filters from 32 to 16 with no loss of accuracy. This improvement can save 3ms time and 10 million multiplication and addition operations. For *nonlinearities part*, in order to be able to use swish function on mobile devices and reduce its computational overhead, hard-swish is proposed in this paper. The formular is $h-swish[x] = x\frac{ReLU6(x+3)}{6}$. The authors believe that H-SWISH has many significant advantages without significantly affecting the model accuracy: almost all software and hardware frameworks provide optimized implementation of ReLU6; In quantization mode, the accuracy error caused by Sigmoid in different implementations is reduced. Even with the quantized version of Sigmoid, it tends to be slower than ReLU. In this experiment, h-Swish in quantized mode was 15% faster than swish in original mode.

In detection experiment, author use MobileNetV3 as a drop-in replacement for the backbone feature extractor in SSDLite and compare with other backbone networks on COCO dataset. MobileNetV3-Large is 25% faster than MobileNetV2 with near identical mAP. MobileNetV3- Small with channel reduction is also 2.4 and 0.5 mAP higher than MobileNetV2 and MnasNet at similar latency. For both MobileNetV3 models the channel reduction trick contributes to approximately 15% latency reduction with no mAP loss.

Summary of work was done

1. I have collected 70,000 images containing human faces with and without mask by downloading public dataset and crawling from the website.
2. I have classified most of the images into with and without mask
3. I have studied the computer vision basic by watching CS231n and finished its coursework. As well as the OpenCV. In trained a Haar cascade to detect mask to detect a face mask using 50x50 box. But after hours of training, its result disappointed me by its high delay that only when I stay still It could perform well.
4. I made a minimal PyTorch implementation of YOLOv3, with support for training, inference, and evaluation. By doing it, I am familiar with the basic of Pytorch.

Problems was faced

1. Labeling such a great number of images is difficult and would cost great time and efforts.
2. MobileNet has new vision which perform better. I need to try the new structure to improve my project so that I can get better accuracy and lower delay.

Solutions were found

1. Utilize public pretrained model to process the images and label them, and then check the result manually to label them correctly. 2. Comparing with MobilNetV2, I found that MobileNetV3 is more powerful for the project, so how to design a convolutional neural network by MobileNetV3 will be a great job. The next immediate steps 1. Study MobileNetV3 and reiterate a simple project. 2. Research some trained mask detector model to label my dataset
是否符合进度？On schedule as per GANTT chart? [YES/NO] YES
下一步 Next steps: 1. Design and implemented the detector to identify human and then human face, which will bring great accuracy for later detection of mask as the paper showed. 2. Using Flask and Vue.js as backend to implement a small website, it will contain Api for later model. And the website will have the information of number of people detected, number of people with and without masks, and an alarm function to notify people without mask.

- Mid-term Progress Report

北京邮电大学 本科毕业设计（论文）中期进度报告
Project Mid-term Progress Report

学院 School	International School	**专业 Programme**	**Telecommunications Engineering with Management**		
姓 Family name	Yang	**名 First Name**	Yongqi		
BUPT 学号 BUPT number	20182121807	**QM 学号 QM number**	190015176	**班级 Class**	2018215108
论文题目 Project Title	Facial Mask Wearing Identification System Based on Deep Learning				

是否完成任务书中所定的中期目标？Targets met (as set in the Specification)?
[YES/NO] YES

已完成工作 Finished work:

1. Paper about the project I recently read:

Focal Loss for Dense Object Detection

This paper is about the RetinaNet. RetinaNet is an architecture that integrates deep residual learning (ResNet) with Feature Pyramid Network (FPN). In this paper, the most important part is its loss function: Focal loss. It is an improvement based on cross entropy, which multiply $(1-p_t)^\gamma$ to adjust the contribution of different probability sample to loss. For a certain sample, a larger probability indicates a greater probability of correct classification. It is natural that we should assign a smaller loss value to this sample, so that the parameters can be updated in the direction of reducing the Loss value of the samples with lower confidence probability when the parameters are backpropagated and updated. In other words, when using Focal Loss, try to update parameters according to the loss calculated for samples that are difficult to classify, and basically make no contribution to Loss for those samples that are well classified.

$$FL(p_t) = -\alpha(1-p_t)^\gamma \log(p_t)$$

RetinaNet is a unified target detection network consisting of one backbone and two subnetworks. The main function of backbone is to obtain the feature map of the whole input image through a series of convolution operations. Target classification and location regression are performed for the two subnets based on the output-feature map respectively. RetinaNet convolution uses ResNet, upsampling and side connection is still FPN compared to original FPN. Through the backbone network, a multi-scale characteristic pyramid is generated. Then two subnets are connected for classification and regression respectively. Overall, the network structure is very simple. The focus of the author is not to innovate the network structure, but to verify the effectiveness of Focal Loss.

2. Finished the dataset collection

In my project, I have three training parts to get a robust mask detection. So, I chose different datasets for training, including COCO dataset for human subject detector,

PubFig which consists of 58,797 images of 200 human for face detector. Besides, I have downloaded the Real-World Masked Face Dataset which ensure there are almost 80,000 classified images for mask detection.

3. Completed human face detection.

To reduce time and hardware costs, I utilized transfer learning on the pretraining model of RetinaNet. Since the COCO dataset is large for RetinaNet training, considering my GPU recourse is not enough for training a RetinaNet from the beginning, which may cost more than one month, I chose a pretrained model from GitHub and used that for further training. I modify the last layer of pretrained so that it can detect human correctly.

After that I apply the MTCNN classifier on the instances in the dataset: CelebA, and label images into “face” and “no face”. And it got an accuracy of 94%

4. Implemented a web app to take real-time photo for mask detection

Till now, I implemented the web app based on Flask and Vue.js and Bootstrap. Firstly, I used Flask to serve a backbone of the web. I implemented an interface for counting numbers of masked and unmasked. And class for picture transferring, which can take the uploaded picture to the module. Besides, I implemented a function that can count the number of human faces, number of humans with mask and without mask. These data can be used for further visualization.

尚需完成的任务 Work to do:

1. Train a robust model to for mask detection and the accuracy should over 95%
2. Optimize the webpage so that it can take the video and visualize the pie chart of people with mask and without mask.
3. Reiteration some other mask detection. Doing this aims to get different result to compare, so that the outstanding point of my project can be showed better.
4. Complete the final report.

存在问题 Problems:

Till now, the model can detect face correctly on images, but when it works not good enough for video. Considering the practical use, I need to improve its performance on video.

拟采取的办法 Solutions:

Considering the time cost in my project, there are 2 parts I can reduce it: video transferring and model processing. For video transferring, I consider choosing fewer frames of the video, and then reduce their resolution to transfer to the model. It can reduce a lot of transfer costs. For model processing, I can choose a larger CPU or take a multi-thread method to reduce it .

论文结构 Structure of the final report:

1. **Abstract**
   Briefly introduce the background and the technique used in the project. Illustrate the feature of project and show its performance.
2. **Background**
   Introduce the global situation of COVID-19 and the importance of wearing masks, as well as the costs of company to remind customer wearing a mask
   Introduce the features and functions of RetinaNet, mainly its
   Introduce the MobileNetV3
   Introduce related research about mask detection
3. **Design and Implementation**
   (1) Design
      Use pipeline flow to introduce the whole system
      Introduce the pipeline of mask detection
      Introduce the layer construction of the network
   (2) Implementation
      Introduce the composition of dataset
      Introduce the human subject detection part
      Introduce the human face detection part
      Introduce the face mask detection part
      Introduce the implementation of webpage
4. **Results and Discussion**
   Analyze the result from accuracy, precision, recall, loss and time complexity
   Compare the results with the other relevant research about mask detection

5. **Conclusion and Further work**
6. **Reference**
7. **Acknowledgement**
8. **Appendix**
9. **Risk and environmental assessment**

- Supervision Log

北京邮电大学 本科毕业设计（论文）教师指导记录表

Project Supervision Log

学院 School	International School	**专业 Programme**	**Telecommunications Engineering with Management**		
姓 Family name	Yang	**名 First Name**	Yongqi		
BUPT 学号 BUPT number	2018212807	**QM 学号 QM number**	190015176	**班级 Class**	2018215108
论文题目 Project Title	Facial Mask Wearing Identification System Based on Deep Learning				

Please record supervision log using the format below:

Date: dd-mm-yyyy
Supervision type: face-to-face meeting/online meeting/email/other (please specify)
Summary:

Date: 17-10-2021
Supervision type: email
Summary: discussed the composition of dataset and method and instructed me how to collect it.

Date: 30-10-2021
Supervision type: online meeting
Summary: feedback the dataset situation, research on the implementation methods

Date: 15-11-2021
Supervision type: online meeting
Summary: discussed the whole design of project and determined the neuron network would be used

Date: 22-11-2021
Supervision type: online meeting
Summary: checked and modified the specification

Date: 09-01-2022
Supervision type: online meeting
Summary: checked and discussed the progress, made suggestions on face detection. Check the early-term report.

Date: 05-02-2022
Supervision type: online meeting
Summary: proposed an open-source project that related to my project,

Date: 11-02-2022
Supervision type: online meeting
Summary: check the project progress

Risk and environmental impact assessment

Risk	Likelyhood	Consequnce	Score	Actions
Computer system crashes	2-Unlikely	1-Minor	2-Low Risk	Save files in a timely manner.
Insufficient computer storage space	3-Moderate	2-Serious	6-Moderate Risk	Delete some unimportant contents.
Doesn't has GPU or GPU is not efficient	3-Unlikey	4-major	12-Significant Risk	Choose a computer with 3080 ti
Infringe someone's copyright.	0-Impossible	4-Major	0-Low Risk	Nothing like this happened, be very careful.
Some electrical resources may be required to complete the project	5-Expected to happen	0-Negligible	0-Low Risk	It seems like a must waste, but we can cut off the power when we are not using devices.
Potential loss to the other individuals or organisations.	0-Impossible	0-Negligible	0-Low Risk	It is not harmful.