DETECTING FACES WITH COVID PROTECTION MASKS FROM IMAGES SHOT IN PUBLIC PLACES USING NEURAL NETWORKS

Mr. Hangkai Wang

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science and Information Technology
Department of Mathematics and Computer Science
FACULTY OF SCIENCE
Chulalongkorn University
Academic Year 2020

การตรวจจับใบหน้าที่ใส่หน้ากากป้องกันโรคโควิดจากภาพที่ถ่ายในสถานที่สาธารณะโดยใช้โครงข่ายประสาท

นายหางไค หวัง

นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ภาควิชาคณิตศาสตร์และ
วิทยาการ คอมพิวเตอร์
คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563

Thesis Title DETECTING FACES WITH COVID PROTECTION MASKS FROM IMAGES SHOT IN PUBLIC PLACES USING NEURAL NETWORKS
By Mr. Hangkai Wang
Field of Study Computer Science and Information Technology
Thesis Advisor Professor CHIDCHANOK LURSINSAP, Ph.D.

Accepted by the Faculty of Science, Chulalongkorn University in Partial Fulfillment of the Requirements for the Master's Degree

Dean of the Faculty of Science
. . . .
(Professor POLKIT SANGVANICH, Ph.D.)

THESIS COMMITTEE

. . . . Chairman
(Associate Professor SUPHAKANT PHIMOLTARES, Ph.D.)

. . . . Thesis Advisor
(Professor CHIDCHANOK LURSINSAP, Ph.D.)

. . . . External Examiner
(Prem Junsawang, Ph.D.)

หางไค หวัง: การตรวจจับใบหน้าที่ใส่หน้ากากป้องกันโรคโควิดจากภาพที่ถ่ายในสถานที่สาธารณะโดยใช้โครงข่ายประสาท. (DETECTING FACES WITH COVID PROTECTION MASKS FROM IMAGES SHOT IN PUBLIC PLACES USING NEURAL NETWORKS) อ.ที่ปรึกษาหลัก : ศ. ดร. ชิดชนก เหลือสินทรัพย์, 50 หน้า.

ตั้งแต่ปี 2019 Covid-19 กลายเป็นปัญหาทั่วไปที่ส่งผลกระทบต่อมวลมนุษยชาติ โรคนี้แพร่กระจายไปทั่วโลกได้สำเร็จ การสวมหน้ากากอนามัยสามารถป้องกันการติดเชื้อได้จริง ดังนั้นการตรวจจับคนที่สวมและไม่สวมหน้ากากในที่สาธารณะจึงเป็นสิ่งสำคัญ อย่างไรก็ตาม ยังมีพื้นที่บางส่วนที่จะปรับปรุงความแม่นยำในการตรวจจับของวิธีการปัจจุบัน ในเอกสารนี้ใช้แบบจำลองการเรียนรู้แบบถ่ายโอนและแบบจำลอง FR-TSVM เพื่อศึกษาข้อมูลล่าสุดของสถานการณ์การแพร่ระบาดของโรคปอดบวมในโควิด -19 ประการแรกชุดข้อมูลของภาพใบหน้าความละเอียดสูง 11,600 ภาพที่สวมหน้ากากและไม่สวมหน้ากากในที่สาธารณะถูกรวบรวมเพื่อการฝึกอบรมการทดสอบและการตรวจสอบความถูกต้อง รูปภาพจะถูกใส่ลงในโมเดล VGG ที่ปรับปรุงแล้ว จากนั้นโครงสร้างของโมเดล VGG ถูกใช้เพื่อดึงคุณสมบัติของรูปภาพ คุณสมบัติเหล่านี้ได้รับการฝึกฝนโดย FR-TSVM พร้อมแนวคิดที่คลุมเครือ วิธีนี้สามารถบรรลุความแม่นยำ 95.125% และยังสูงกว่าผลการตรวจจับของวิธีการอื่น ๆอื่น ๆ

สาขาวิชา วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ ลายมือชื่อนิสิต ....

ปีการศึกษา 2563 ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6278015323: MAJOR COMPUTER SCIENCE AND INFORMATION TECHNOLOGY

KEYWORDS: FACIAL IMAGES / PUBLIC / MASK / VGG / FR-TSVM

HANGKAI WANG : DETECTING FACES WITH COVID PROTECTION MASKS FROM IMAGES SHOT IN PUBLIC PLACES USING NEURAL NETWORKS. ADVISOR : PROF. CHIDCHANOK LURSINSAP, Ph.D., 50 pp.

Since 2019, Covid-19 has become a common problem affecting all mankind. The disease has successfully spread all over the world. Wearing a mask can practically protect the infection. Thus, detecting people wearing and not wearing masks in public is essential. However, there is still some room to improve detection accuracy of the present methods. In this paper, the transfer learning model and FR-TSVM model are used to study the latest data of pneumonia epidemic situation in Covid-19. First, a data set of 11600 facial images wearing masks and not wearing masks in public was collected for training, testing, and validation. The pictures will be put into the improved VGG model. Then the structure of VGG model was used to extract the features of images. These features were trained by FR-TSVM with fuzzy concept included. This approach can achieve 95.125% accuracy and it is also higher than the detection results of other methods.

Field of Study:	Computer Science and Information Technology	Student's Signature	....
Academic Year:	2020	Advisor's Signature	....

Acknowledgements

First, I would like to thank my consultant Professor Dr. Chidchanok Lursinsap. I have been grateful to my teacher for his help in my thesis. Because the teacher usually has other students to guide, but my thesis teacher will patiently answer every question, and give me a lot of advice on the topic selection and ideas of the thesis.

Secondly, I would like to thank the other members of dissertation committee: Associate Professor Dr. Suphakant Phimoltares and Dr. Prem Junsawang. The teachers also gave relevant opinions in my thesis writing, and gave relevant suggestions on some common mistakes in the thesis.

At the same time, I am also appreciating my family and friends, who have been by my side to encourage me to improve myself and challenge myself.

I am very grateful to my advisor for his help and encouragement during my thesis. I am also very grateful to all teachers for your sincere suggestions and guidance during the thesis defense. Thank you.

Hangkai Wang

CONTENTS

Page

Abstract (Thai) . . . . **iv**

Abstract (English) . . . . **v**

Acknowledgements . . . . **vi**

Contents . . . . **vii**

List of Tables . . . . **ix**

List of Figures . . . . **x**

1 Introduction **1**

1.1 Overview 1

1.2 Survey previous work 2

1.3 Research Approach 5

2 Background **7**

2.1 VGG 7

2.2 FR-TSVM 8

2.3 Other Methods 9

3 PROPOSED METHOD **11**

3.1 Model Framework 12

3.2 Model Design 13

4 Experiment **21**

4.1 Data Set 21

4.2 Experimental Setup 23

4.3 FR-TSVM Part 30

4.4 Output of Combined Model 32

Page

4.5 Comparative Other Methods 34

5 Analysis 39

6 Discussion and Conclusion 42

6.1 Summary of Findings 42

6.2 Future Work 43

References 45

Appendix 48

Apppendix A Appendix 48

A.1 Features 48

Apppendix B List of Publications 49

B.1 International Conference Proceeding 49

Biography 50

LIST OF TABLES

Table **Page**
4.1 THE TRAIN LOSS AND VAL LOSS 32
4.2 RESULTS FROM FR-TSVM 34
4.3 COMPARATIVE RESULTS 37
5.1 COMPARISON OF CONFUSION MATRIX 40
5.2 COMPARATIVE OF PROS AND CONS 40

LIST OF FIGURES

Figure **Page**

3.1 The framework of the proposed Model. 12
3.2 Example of pixel change of one face image in VGG. 14
3.3 One example of the output of VGG model. 16
3.4 The example of check correct number of arguments in FR-TSVM. 18
4.1 Screenshot from the MAFA dataset on the kaggle website. 21
4.2 Screenshot from the WIDER dataset on the kaggle website. 22
4.3 Some examples of facial images with masks in face image data set. . . . 22
4.4 Install the all needed libraries. 24
4.5 Define the path for the face image. 25
4.6 One random face image in the train folder. 27
4.7 Image processing code. 27
4.8 An example of a fine-tuned image. A woman wears an N95 mask. 28
4.9 Training images by VGG model. 28
4.10 The change of specific image pixels in each layer of the VGG model. . . 29
4.11 The code for extracting features. 29
4.12 Using FR-TSVM model to get result. 31
4.13 Outclass in FR-TSVM model. 31
4.14 ValY in FR-TSVM model 32
4.15 Loss-curve in this model. 33
4.16 The accuracy curve in this model. 33
4.17 Difficult images to detect by proposed method. 34
4.18 The results from using YOLOv3. 36
4.19 The results from using YOLOv3 based on parameters. 36
A.1 Features extracted from the VGG model 48

Chapter I

INTRODUCTION

1.1 Overview

Since COVID-19 broke out, it has not ended yet. To a certain extent, the prediction of COVID-19 epidemic can help us master more information [1,2]. People wearing masks in public has been proved to be effective by WHO [3], and a study has also proved that it is a cost-effective method [4]. Although wearing masks has proved to be effective, we cannot guarantee that everyone will wear masks consciously in public, so we need some machines to help us detect whether people wear masks in public. Detecting whether people wear masks in public places can also share the pressure of the government and save human resources. That's why it's important to test people wearing masks.

In this paper, a new face detection method based on double support vector machines [5, 6] is discussed experimentally, which is a model combining VGG and FR-TSVM [7].

The advantage of using SVM in face image experiment is that it doesn't need to use all the face image data, but only select some of them, which is suitable for small sample face image classification. However, SVM is not suitable for face image classification with large samples.

If we want to use large sample face image classification, we can consider using FR-TSVM. Compared with traditional SVM, FR-TSVM is faster and has better training effect. The biggest disadvantage of FR-TSVM is that it can't learn the representation of face image features automatically, and the training effect of FR-TSVM is affected by parameters in classification experiments.

1.2 Survey previous work

According to some studies on the case data of COVID-19 epidemic [8], we found that the virus can also spread from person to person [9].

COVID-19 is widely spread all over the world, and there are many studies on how COVID-19 virus caused the pandemic [10,11], including many studies on the relationship between COVID-19 and masks [12]. In many experiments to test whether people wear masks, different models are usually compared to obtain test results.

In fact, there have been a lot of experiments on face recognition, not only in this paper. The methods are also varied.

For example, in paper[13], the author uses YOLO model to detect masks, obtains different accuracy rates by comparing the models, and obtains the performance of the model according to the final accuracy results. YOLO algorithm is very fast and used for target detection here, which can not only identify masks or other objects in different face images, but also obtain the sizes and positions of different objects in face images and express them in the form of coordinates.

YOLO algorithm also has some shortcomings. YOLO algorithm uses windows to detect mask parts in face images. If it is necessary to detect objects of different sizes in face images, many windows of different sizes are needed. YOLO's recognition of different objects in face images will greatly increase the amount of calculation.

Theoretically, the face mask types that YOLO can detect are related to the trained face samples, and the size of the face images input to the model is fixed.

When there are enough face images trained, YOLO can detect all kinds of face images. YOLO model can also improve the accuracy after some improvements. In paper [13], YOLO v3 is used to detect whether a face is wearing a mask or not.

By optimizing YOLO v3 model, the accuracy is from 85.1% to 86.3%. In order to achieve higher accuracy, the author can actually use the latest YOLO v5 algorithm to detect face images, increase trained face images and optimize YOLO v5 models.

In paper [14], the author uses M-CNN model to detect masks, and finally detects masks with an accuracy rate of 91.5%. The advantages of M-CNN are that it has few parameters, small vector dimension and little computation.

But the author focuses on the training of face images with one kind of masks, and finally, the types of masks that can be identified are single, mainly medical masks, while other types of masks are difficult to identify. Here, the author only used 1200 face images for training and 300 for testing. The size of input face images are 150*150, and the size is fixed and cannot be changed. Moreover, the M-CNN model used by the author is very simple, just changing the CNN model. Although the final accuracy rate is 91.5%, this way only suitable for small face image samples and single face mask images are used for recognition.

In fact, the author can increase the training sample of face images to more than 10,000 face images, which should include various types of face mask images. And with the increase of face image sample, M-CNN model can be more optimized to increase the final accuracy.

From these studies, we can spread our ideas and test whether people wear masks in public places from different angles. For example, in some VGG experiments, better detection results can be obtained by improving the efficiency of the model [15-17].

If the VGG model is used alone to detect face images as a classifier, the advantage is that the structure of VGG can be changed. The structure of VGG is also very simple. The disadvantage is that if VGG is not suitable for large face image samples, the calculated parameters will be very large, occupying a very large memory and wasting computing resources.

The size of face images input into the VGG model needs to be fixed, which is related to the fully connected layer in the VGG. The types of masks that VGG can recognize are related to the trained face samples. If VGG only trains a small sample of face pictures with a single mask, then only one kind of face mask images and face images without a mask can be recognized in the end. In theory, when VGG can train a large number of face image samples, including various kinds of masks, it can also detect various kinds of masks when finally detecting.

If we only consider using VGG to detect face images, it is necessary to use more than 10,000 face images for training. At the same time, we can consider optimizing VGG model, such as removing the three full connection layers of VGG and using image augmentation to improve the accuracy of detecting face images by VGG.

The experiment to be done in this paper is to use the deep transfer learning model to identify whether people wear masks or not in public places, not only VGG, but FR-TSVM model.

Because of COVID-19, many people who go out begin to wear masks. This paper considers an innovated model. In addition, people who go out wear a variety of masks, so whether the model can be used to identify various masks in this experiment. During COVID-19, people may appear in pairs in public places. This experiment hopes to improve the model, which can not only judge whether one person wears a mask, but also judge whether many people wear a mask at the same time.

This paper will innovative use the improved VGG model and FR-TSVM model to detect whether people wear masks or not. According to the final experimental results, it is found that this is an efficient way to detect various types of face images. Moreover, this proposed model is very fast and occupies less memory.

The innovative model in this paper can detect various masks on the basis of training 10000 face images. The input face images can include various sizes, and this proposed model can detect whether many people wear masks at the same time.

However, there are some defects in this experiment, such as it is difficult to adjust the parameters in the experimental model to obtain higher accuracy. At the same time, some special types of ”masks” can not be identified by the model of this experiment. When the mouth part of some face images is blocked by objects, such as fans, cups or fruits, this proposed model can not correctly identify such face images. The classification standard of data sets is simply based on whether to wear masks or not.

These are the places that need to be changed in future experiments. The VGG model can be further optimized, and a large number of experiments can be done to adjust the parameters of nonlinear FR-TSVM to obtain better model performance.

1.3 Research Approach

The research methods of this paper are mainly divided into the following four steps.

- Collect effective data sets: many high-definition images of people wearing or not wearing masks in public.
- Use VGG model to process pictures.
- The processing results of face images are input into FR-TSVM model.
- Use the experimental results to calculate the corresponding accuracy and other indicators.

The model in this experiment uses VGG model to extract features, and then uses FR-TSVM to judge whether people wear masks in public. The model can detect large samples, and the method has the characteristics of fast learning speed, less computation, high accuracy and good robustness of the model. At the same time, the method is suitable to detect various face masks, and the accuracy rate is as high as 95.125%. In addition, this method can also detect multiple faces with masks in the image.

After the model is combined with an artificial intelligence machine, the machine can be used to detect whether people wear masks in public. This model can also be used for further research of face detection experiments.

Until today, COVID-19 is still raging, and we hope this model can help relevant organizations and governments to carry out face detection.

Chapter II

BACKGROUND

There are various types of masks and scenes of the input images. This complex data sets make it difficult to extract the appropriate features for classifying wearing mask and not wearing mask images. The proper approach is to deploy the method of convolution neural network where the process of feature extraction is included as a part of learning process. Three main relevant concepts, i.e. VGG, FR-TSVM, and YOLO, are briefly summarized in this chapter.

2.1 VGG

The Visual Geometry Group (VGG) Network has two main structures of different depth, namely VGG16 and VGG19. VGG can be conveniently used for migration learning. VGG was specifically designed to extract image features and classify the features by employing the structure of multi-layer neural network. Both operations are combined into one single network. VGG has been applied to several areas whose input data are in the domain of images. The depth of VGG depends upon the how good the features are extracted so that the overlap of classes is minimum. One complex example of VGG applications is radar target recognition [18]. This application proposed a modified version of VGG with different number of layers and also an optimizer called adamax. The concept of regularity was implemented to prevent over-fitting problem. The number of layers of VGG is not fixed for all applications [19,20,21,22]. For example, there are 13 convolution layers and 3 fully connected layers in the Visual Geometry Group Network structure [19]. Convolution layer and fully connected layer both have weight coefficients called weight layers. In VGG16, the convolution kernel size is (3,3), and the pool kernel size is (2,2). The architecture of VGG16 is composed of three consecutive convolution layers and one pooling layer. This architecture of VGG16 can reduce the number of weights and speed up the computation. Furthermore, the reduction of number of weights can prevent over-fitting problem. The output value of each neuron in each convolution layer is computed by an activation function which can fit more complex data.

2.2 FR-TSVM

Support vector machine (SVM) was designed to cope with linearly separable data sets when they are mapped onto another higher dimensional space by using a kernel function. SVM does not need any training but it uses the concept of Lagrange's multipliers with a cost function to find the location of the optimal hyperplane. The time complexity of SVM to compute the weights of the optimal hyperplane is $O(n^3)$, where n is the number of training data. This implies that the learning time is controllable. However, if the size of a training set is very large, SVM is not suitable because it must inverse a matrix of size n^2, which may cause the memory overflow.

TSVM (twin SVM) uses more than one hyperplane to separate the data. This approach can improve the accuracy of separability. To further improve TSVM concept, fast and robust TSVM (FR-TSVM) was proposed by including the concept of fuzzy membership into the cost function. The added term makes the location of separating hyperplane more flexible to bias the optimal location when being applied to the testing data. The shortcoming of FR-TSVM depends on the reference object. Compared with deep learning, its biggest disadvantage is that it can not automatically learn the representation of features; Compared with Softmax multi-classification, FR-TSVM needs to construct multiple secondary classifications and use voting mechanism to do multi-classification, which is not concise enough. In this theis, FR-TSVM is used as the final classifier after all relevant features of the training images are extracted by using a convolution neural network.

2.3 Other Methods

At present, there are many ways to detect masks, such as paper [13], which focus on YOLO (You Only Look Once) v3 algorithm in face mask experiment. YOLO algorithm can be used for target detection. Different face images have different coordinate axes. In YOLO algorithm, the face mask image can be regarded as a two-dimensional matrix. After a lot of training in face mask images, YOLO can infer the approximate position of the mask corresponding to the tested face picture through the partial image information with or without masks. It can distinguish the background from the mask part to be detected, and YOLO algorithm uses a matrix box to frame the face target to be detected.

YOLO v3 model is much more complicated than YOLO v1 and YOLO v2. The speed and accuracy can be weighed by changing the size of the model structure. We apply the YOLO v3 model to face mask data set to get detection results. YOLO uses a convolution neural network (CNN) to train the face mask data set, and then uses full connection layer to get the predicted results. The initial parameter setting can be pre-trained on ImageNet.

In paper [14], which uses a convolution neural network to detect face masks. The author uses the improved CNN model to detect a large number of face mask pictures. CNN extracts features for the whole face picture, while M-CNN can focus on whether there is a mask or not. The parameters are pre-trained on ImageNet classification.

This thesis uses VGG and FR-TSVM models to detect that whether a face wears a mask or not. All images are resized to 32*32*3 at the beginning. The size of each image changes constantly after each layer of VGG processing. After we train a lot of processed images on VGG model, we can group them by features of wearing masks and not wearing masks, and group pixels representing these features of wearing masks and not wearing masks. The last layer of VGG model is extracted to generate the corresponding file, and the feature vector needed is extracted. The extracted face mask or without mask feature vector is fed into FR-TSVM model and get the final output.

Chapter III

PROPOSED METHOD

VGG model is one of the easiest and convenient neural network models for image classification [23,24]. But its structure is rather complicated due to the designed functions. The original structure of VGG16 model was designed to combine two functional networks working in a sequential order. The first network performs the automatic image feature extraction as well as encoding the features. These features are passed to the second network to classify them according to the training targets. The size of VGG16 is rather large due to many layers and neurons in each layer. To control the training time and accuracy of a VGG16, some parameters of VGG16 must be tentatively adjusted. The number of parameters are not so few and difficult to adjust. To ease the training process and to improve the speed of training, the classifying structure is replaced with a SVM structure and the feature extracting structure of VGG16 is still maintained. In this thesis, a special fast SVM called FR-SVM was employed instead of a typical SVM to improve the optimization speed of classification and also retain the high accuracy. Thus, the last three layers of VGG16 was removed and a FR-SVM was plugged into the network. The weights in VGG16 model will be automatically downloaded when the VGG16 model is initialized, and the weights pre-trained on imageNet dataset were used here.

3.1 Model Framework

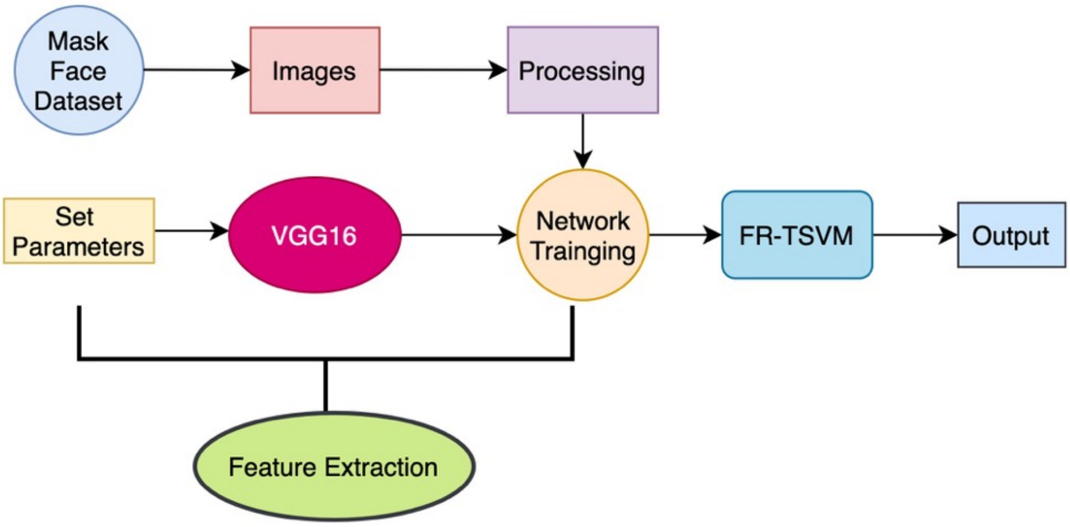


Figure 3.1: The framework of the proposed Model.

The proposed frame is shown in Fig. 3.1. A given image may contain many faces. Each face must be extracted first. Since the main concern of this thesis focuses on the frame work of recognizing a face image with medical mask, extracting a face from a given image can be done by using any existing tools. Two data sets from WIDER and MAFA were used for training ans testing. After extracting faces from image data sets, VGG16 neural network is deployed to extract the features of face training and testing set. To do this, the top three layers of VGG16 are removed. Then, the output from the top layer are used as the features of each image. The features of training images from VGG16 are fed to a FR-TSVM classifier. The trained VGG16 is also used for extracting the features of testing data. By using FR-TSVM for classification, the time for training VGG16 can be speedup because the structure of VGG16 in our framework is smaller than the original VGG16 structure. Furthermore, the degree of overfitting after training may be reduced.

3.2 Model Design

Data Set Information

The selected data sets are from WIDER and MAFA data sets. All face images are in JPG format. In MAFA data set, there are 25,876 train-images, including face images of various masks, such as cloth masks, N95 masks, medical masks, disposable mask and transparent masks. The masks in the data sets appear in various situations, including masks on the faces of crowds and a mask on a single face. The face images are mainly the faces of people in Asia. No faces of people from other continents are in this data set. The MAFA data set also contains images with different facial postures.

For WIDER data set, there are 32,203 face images with various types of face masks. WIDER classifies face images according to very specific scenes. It is partitioned into 62 different types, including parade, handshake, demonstration, riot, dancing, fun, caring, meeting, shoppers and so on. Image types include various races, colors, genders and ages. It also includes images of a single person and multiple people, as well as pictures of people and animals. There are unusual 4,953 face images from WIDER data set, including mask images of all ages, and face images that are difficult to recognize such as face images with hands covering the face.

For the training set, 5000 images with masks and 5000 images without masks were randomly selected from both data sets. Similarly, 1600 images were also randomly selected for validation and testing sets.

VGG-16

In this experiment, 10,000 face images are used for training, including mask and no mask face images, of which 800 face images are used for validation and 800 face images are used for testing. The experiment of VGG part is conducted in kaggle with TPU v3-8 and COLAB with GPU on a Mac with an Inter Core i9 processor

2.3GHz with 16 GB RAM.

A large number of face images will be processed by using VGG model, which is trained in advance. The transferring learning is completed by changing the structure of VGG16. When the VGG model is initialized, the weights will be automatically downloaded. And the weights of VGG16 model are automatically adjusted with the training face mask data set.

VGG is used as a pre-trained model to train a large data set. The pre-trained VGG model is beneficial to us for many reasons. By using a pre-trained model, the training time can be saved, which concerns the reduction of the calculation time and improving the overall efficiency of the face detection experiment.

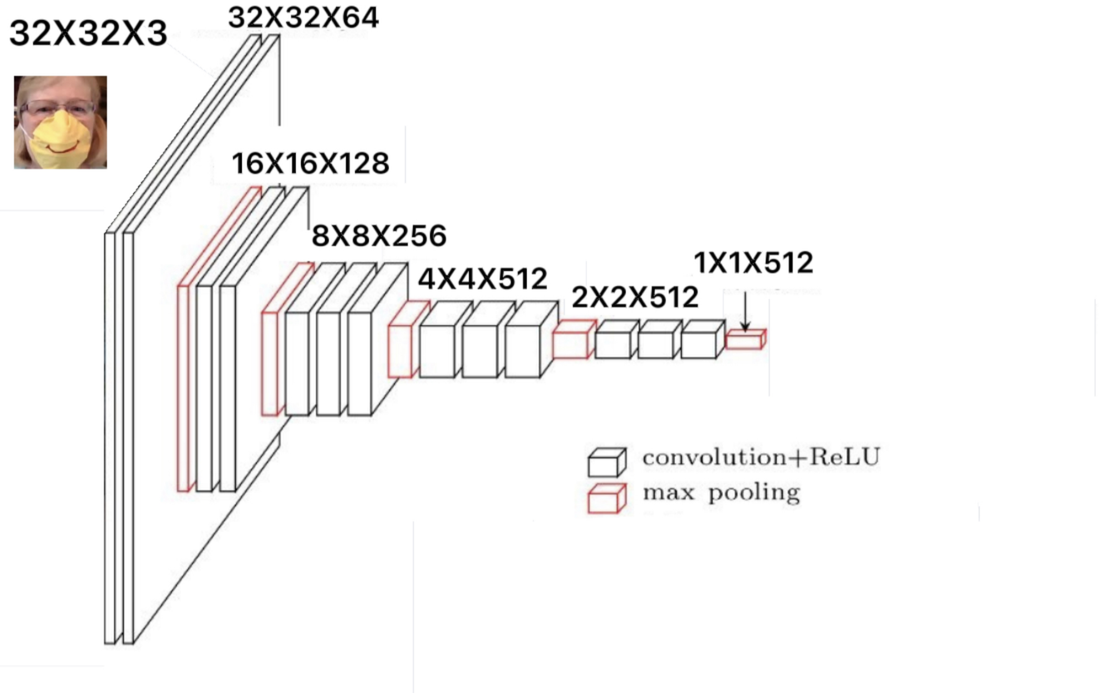


Figure 3.2: Example of pixel change of one face image in VGG.

The structure of VGG16 used in this thesis is shown in Fig. 3.2. After the face image is input, it is processed by 13 convolution layers and five max pooling layers in VGG16 model. Finally, the relevant features are output from the last max pooling layer. The specific steps are as follows.

- Enter the processed face image in the VGG model with the size of 32*32, and enter the first and second convolution layers. Face image size changed from 32*32*64 to 16*16*128. Then the result is output to the max pooling layer.

- The face mask results output by the maximum pooling layer above are processed by the third and fourth convolution layers using the filter 3×3, face image size changed from 16*16*128 to 8*8*256. Then the result is output to the next max pooling layer.

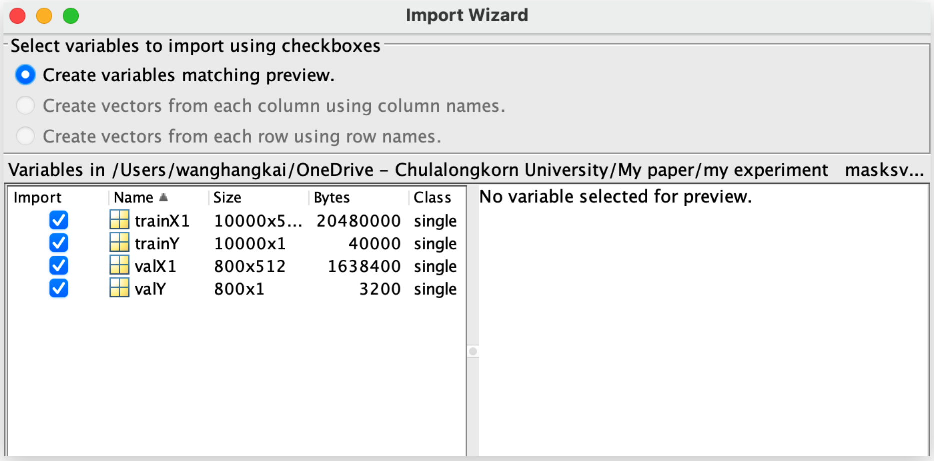
- After the fifth, sixth and seventh layers of processing, the features of the face mask image are input to the next max pooling layer. Face image size changed from 8*8*256 to 4*4*512.

- Then after the eighth, ninth and tenth layers of processing, the features of the face mask image are input to the next max pooling layer. Face image size changed from 4*4*512 to 2*2*512.

- Then after the eighth, ninth and tenth layers of processing, the features of the face mask image are input to the last max pooling layer. Face image size changed from 2*2*512 to 1*1*512.

Then, the new models and weights are saved separately. Finally, we only use VGG 13 layers to get face mask features from face mask data set. No need to use the last 3 fully connected layers.

VGG Output



Import Wizard

Select variables to import using checkboxes

● Create variables matching preview.

○ Create vectors from each column using column names.

○ Create vectors from each row using row names.

Variables in /Users/wanghangkai/OneDrive – Chulalongkorn University/My paper/my experiment masksv...

Import	Name ▲	Size	Bytes	Class
☑	trainX1	10000x5...	20480000	single
☑	trainY	10000x1	40000	single
☑	valX1	800x512	1638400	single
☑	valY	800x1	3200	single

No variable selected for preview.

Figure 3.3: One example of the output of VGG model.

One example of the output of VGG model, as shown in Fig. 3.3. This example shows that 10,000 trained face images and 800 validation face images are converted into corresponding face feature vectors. The trained and validation face images are converted into corresponding forms of trainX1, train Y, valX1 and valY, and the files are saved as mat files. VGG16 in this thesis was implemented by using python with SciPy package. After extracting features from the VGG model, it can be saved in the corresponding file format. Considering that the extracted features should be put into FR-TSVM model in MATLAB, the extracted feature files can be saved in mat file format.

FR-TSVM

The experiment of FR-TSVM part is conducted in MATLAB(R2021a) on a Mac with an Inter Core i9 processor 2.3GHz with 16 GB RAM. In order to solve the problems related to dual form, the optimization toolbox is used in Matlab [25]. This is also why we should consider using different platforms for experiments, because MATLAB itself has some advantages where the other platforms cannot match. The use of some toolboxes on MATLAB can help us draw relevant graphs and export relevant result variables.

A dual coordinate descent, CD method, is used in FR-TSVM. In FR-TSVM, CD method can solve the problem of dual, and CD can be used to calculate the number of iterations.

FR-TSVM model is divided into linear FR-TSVM model and nonlinear FR-TSVM model. The main difference between linear and nonlinear FR-TSVM is the choice of classification hyperplane. Linear FR-TSVM chooses linear hyperplane, while kernel function used in nonlinear FR-TSVM uses nonlinear hyperplane. The concept here is the same as classical SVM. In the face mask recognition experiment, the linear FR-TSVM model gets better results here.

But theoretically, in general, nonlinear FR-TSVM is better than linear FR-TSVM in identifying whether people wear masks or not. Of course, this is also affected by super-parameters. If the parameters are not properly selected, the performance may be degraded. In the specific face mask recognition experiment, we found that the parameters of the nonlinear FR-TSVM model are difficult to adjust automatically according to the features of the input face images. But the mask recognition rate obtained by adjustment is not high in nonlinear FR-TSVM model.

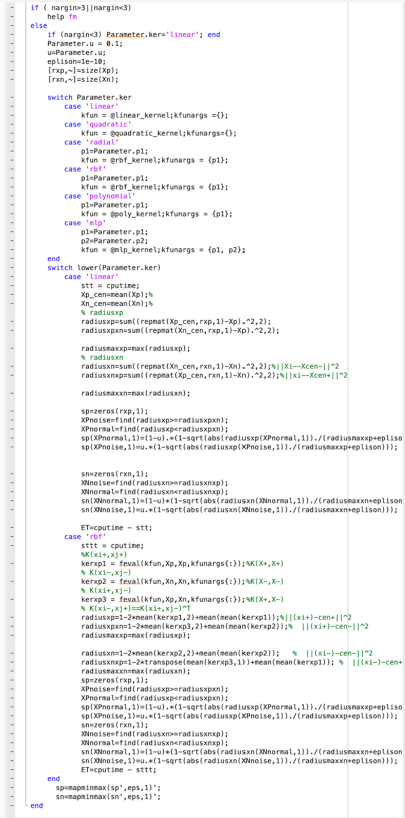
In this thesis, we used a linear model. Linear FR-TSVM can flexibly adjust parameters, transform new information according to the feature vectors of collected face images, and integrate the new information into the established FR-TSVM model.

The test results corresponding to each image in the test set are what we need. The input of FR-TSVM are features extracted using VGG model. The output of the FR-TSVM model is 0(-1) or 1. ”0” means there is a mask, and ”1” means there is no mask. According to the corresponding test results, after running FR-TSVM model, we can calculate the corresponding accuracy, prediction rate, recall, test time and confusion matrix in the face test data set.

Adjust Parameters

There are 6 parameters (i.e., c1, c2, c3, c4, μ, g) in second FR-TSVM model. At the beginning, we use the original parameters of the linear FR-TSVM model in the experiment, and then the parameters are adjusted according to the face mask data set to get a better performance.

The parameters for obtaining the best test results are different in FR-TSVM model according to face mask dataset, the code as shown in Fig. 3.4.



```
if ( nargin>3||nargin<3)
    help fm
else
    if (nargin<3) Parameter.ker='linear'; end
    Parameter.u = 0.1;
    u=Parameter.u;
    eplison=1e-10;
    [rxp,~]=size(Xp);
    [rxn,~]=size(Xn);

    switch Parameter.ker
        case 'linear'
            kfun = @linear_kernel;kfunargs ={};
        case 'quadratic'
            kfun = @quadratic_kernel;kfunargs={};
        case 'radial'
            p1=Parameter.p1;
            kfun = @rbf_kernel;kfunargs = {p1};
        case 'rbf'
            p1=Parameter.p1;
            kfun = @rbf_kernel;kfunargs = {p1};
        case 'polynomial'
            p1=Parameter.p1;
            kfun = @poly_kernel;kfunargs = {p1};
        case 'mlp'
            p1=Parameter.p1;
            p2=Parameter.p2;
            kfun = @mlp_kernel;kfunargs = {p1, p2};
    end
    switch lower(Parameter.ker)
        case 'linear'
            stt = cputime;
            Xp_cen=mean(Xp);%
            Xn_cen=mean(Xn);%
            % radiusxp
            radiusxp=sum((repmat(Xp_cen,rxp,1)-Xp).^2,2);
            radiusxpxn=sum((repmat(Xn_cen,rxp,1)-Xp).^2,2);

            radiusmaxxp=max(radiusxp);
            % radiusxn
            radiusxn=sum((repmat(Xn_cen,rxn,1)-Xn).^2,2);%||Xi--Xcen-||^2
            radiusxnxp=sum((repmat(Xp_cen,rxn,1)-Xn).^2,2);%||xi--Xcen+||^2

            radiusmaxxn=max(radiusxn);

            sp=zeros(rxp,1);
            XPnoise=find(radiusxp>=radiusxpxn);
            XPnormal=find(radiusxp<radiusxpxn);
            sp(XPnormal,1)=(1-u).*(1-sqrt(abs(radiusxp(XPnormal,1))./(radiusmaxxp+epliso
            sp(XPnoise,1)=u.*(1-sqrt(abs(radiusxp(XPnoise,1))./(radiusmaxxp+eplison)));


            sn=zeros(rxn,1);
            XNnoise=find(radiusxn>=radiusxnxp);
            XNnormal=find(radiusxn<radiusxnxp);
            sn(XNnormal,1)=(1-u)*(1-sqrt(abs(radiusxn(XNnormal,1))./(radiusmaxxn+eplison
            sn(XNnoise,1)=u.*(1-sqrt(abs(radiusxn(XNnoise,1))./(radiusmaxxn+eplison)));

            ET=cputime - stt;
        case 'rbf'
            sttt = cputime;
            %K(xi+,xj+)
            kerxp1 = feval(kfun,Xp,Xp,kfunargs{:});%K(X+,X+)
            % K(xi-,xj-)
            kerxp2 = feval(kfun,Xn,Xn,kfunargs{:});%K(X-,X-)
            % K(xi+,xj-)
            kerxp3 = feval(kfun,Xp,Xn,kfunargs{:});%K(X+,X-)
            % K(xi-,xj+)==K(xi+,xj-)^T
            radiusxp=1-2*mean(kerxp1,2)+mean(mean(kerxp1));%||(xi+)-cen+||^2
            radiusxpxn=1-2*mean(kerxp3,2)+mean(mean(kerxp2));%  ||(xi+)-cen-||^2
            radiusmaxxp=max(radiusxp);

            radiusxn=1-2*mean(kerxp2,2)+mean(mean(kerxp2));   %  ||(xi-)-cen-||^2
            radiusxnxp=1-2*transpose(mean(kerxp3,1))+mean(mean(kerxp1)); %  ||(xi-)-cen+
            radiusmaxxn=max(radiusxn);
            sp=zeros(rxp,1);
            XPnoise=find(radiusxp>=radiusxpxn);
            XPnormal=find(radiusxp<radiusxpxn);
            sp(XPnormal,1)=(1-u).*(1-sqrt(abs(radiusxp(XPnormal,1))./(radiusmaxxp+epliso
            sp(XPnoise,1)=u.*(1-sqrt(abs(radiusxp(XPnoise,1))./(radiusmaxxp+eplison)));
            sn=zeros(rxn,1);
            XNnoise=find(radiusxn>=radiusxnxp);
            XNnormal=find(radiusxn<radiusxnxp);
            sn(XNnormal,1)=(1-u)*(1-sqrt(abs(radiusxn(XNnormal,1))./(radiusmaxxn+eplison
            sn(XNnoise,1)=u.*(1-sqrt(abs(radiusxn(XNnoise,1))./(radiusmaxxn+eplison)));
            ET=cputime - sttt;
    end
     sp=mapminmax(sp',eps,1)';
     sn=mapminmax(sn',eps,1)';
end
```

Figure 3.4: The example of check correct number of arguments in FR-TSVM.

From this example, it can be seen that the method of selecting parameters by FR-TSVM is not static. On the contrary, it is very complex, and the parameters depend on the corresponding face data set. In addition, FR-TSVM model can select the corresponding kernel value according to the feature vectors of the input face image. The model can also adjust the parameters to a certain extent according to the feature vectors of the face images.

In the experiment, the parameters of linear TSVM model can be adjusted well, but the parameters of nonlinear FR-TSVM model are difficult to control. The parameters of nonlinear FR-TSVM model have to be adjusted manually. In the future work, the parameter adjustment of nonlinear model is a direction that can be worked hard to get better performance.

Running Time

The running time mainly depends on the size of the face data set and the running environment. In this paper, the time of calculating the whole model is to add the time of getting the experimental results by VGG on kaggle and the time of getting the experimental results by FR-TSVM model on MATLAB. The images were downloaded in advance from kaggle, including 11,600 face images from WIDER and MAFA datasets, 10,000 for training, 800 for testing and 800 for validation.

VGG model was trained with the data from kaggle with TPU v3-8 until face feature vector was successfully extracted and saved in mat file, which took less than 30 seconds. When running the FR-TSVM model in MATLAB, the mat file extracted from VGG was downloaded and saved in advance. The time of running on MATLAB to get the results is less than 0.009 seconds

Reasons Of Our Framework

The whole idea of the experiment is that we will use the first 13 layers of VGG16 model to extract the relevant features of the face mask images, and then use FR-TSVM model to calculate the result of face mask recognition. VGG16 has some shortcomings, such as the large amount of parameters, and most of the parameters are concentrated in the full connection layer. Because we need to train a large number of samples, including 10000 face images, if we use VGG16 to detect face masks completely, it will increase the calculation time and occupy a large amount of computer memory. Furthermore, the adjustment of a large number of parameters can result in poor face detection results.

FR-TSVM also has some shortcomings. Compared with deep learning, the biggest disadvantage of FR-TSVM model is that it can not automatically learn the representation of features. Compared with Softmax multi-classification, FR-TSVM needs to construct multiple secondary classifications to do multi-classification, which is not concise enough. Therefore, replacing the last three fully connected layers in VGG16 with FR-TSVM model is beneficial to the combination of both appraoches, i.e. the advantages of VGG16 and FR-TSVM.

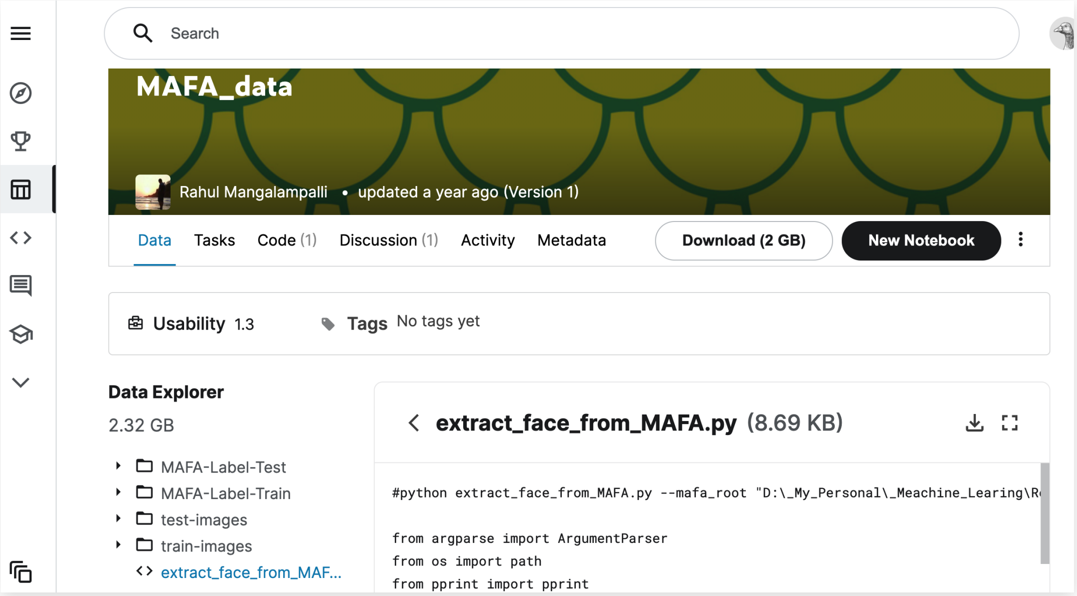
The VGG can use the parameters trained on ImageNet at first, while the parameters of the linear FR-TSVM model can be adjusted to a certain extent according to the face data set. The combination of VGG model and FR-TSVM model is helpful for us to calculate the corresponding face mask detection results. First, only the first 13 layers of VGG model were used, and the series convolution layers have fewer parameters, which can learn the relevant features of face pictures quickly and stably. Secondly, compared with support vector machine, FR-TSVM has the advantages of solving encouraging points, noise, robust and fast speed. FR-TSVM model is also suitable for large face mask images.

Chapter IV

EXPERIMENT

4.1 Data Set

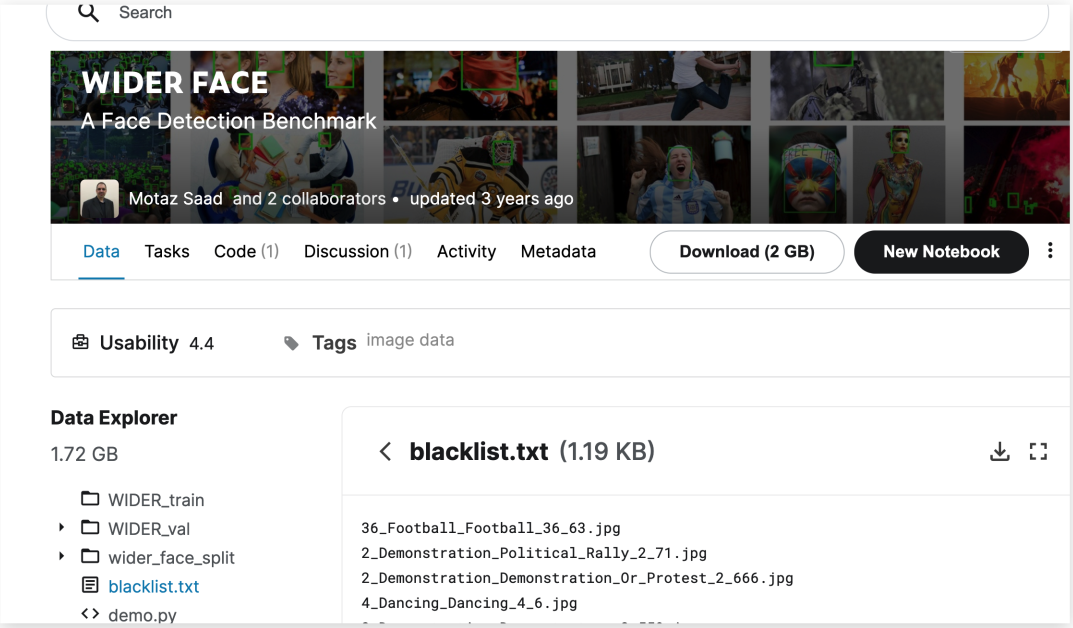
The experimental MAFA data set was downloaded from kaggle, as shown in Fig. 4.1. In MAFA data set, there are 25,876 training images and 4,935 test images. The data set WIDER was also downloaded from kaggle, as shown in Fig. 4.2. There are 32,203 face images in the WIDER data set.



```
#python extract_face_from_MAFA.py --mafa_root "D:\_My_Personal\_Meachine_Learing\R

from argparse import ArgumentParser
from os import path
from pprint import pprint
```

Figure 4.1: Screenshot from the MAFA dataset on the kaggle website.

In MAFA data set, there are many face images with various masks, while in WIDER data set, there are mainly face images in various scenes and many face images without masks. Both of them have rich face images, including various types. Some examples of facial images with masks from WIDER and MAFA datasets are shown in Fig. 4.3



```
36_Football_Football_36_63.jpg
2_Demonstration_Political_Rally_2_71.jpg
2_Demonstration_Demonstration_Or_Protest_2_666.jpg
4_Dancing_Dancing_4_6.jpg
```

Figure 4.2: Screenshot from the WIDER dataset on the kaggle website.

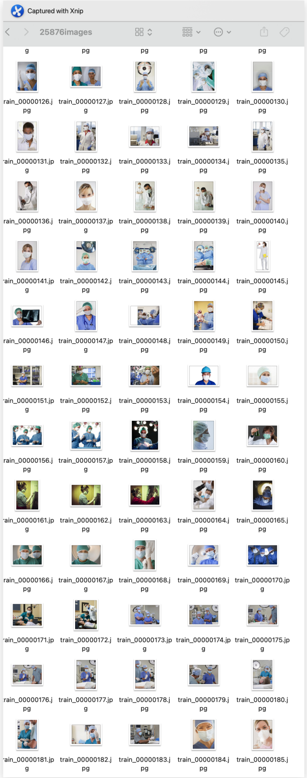
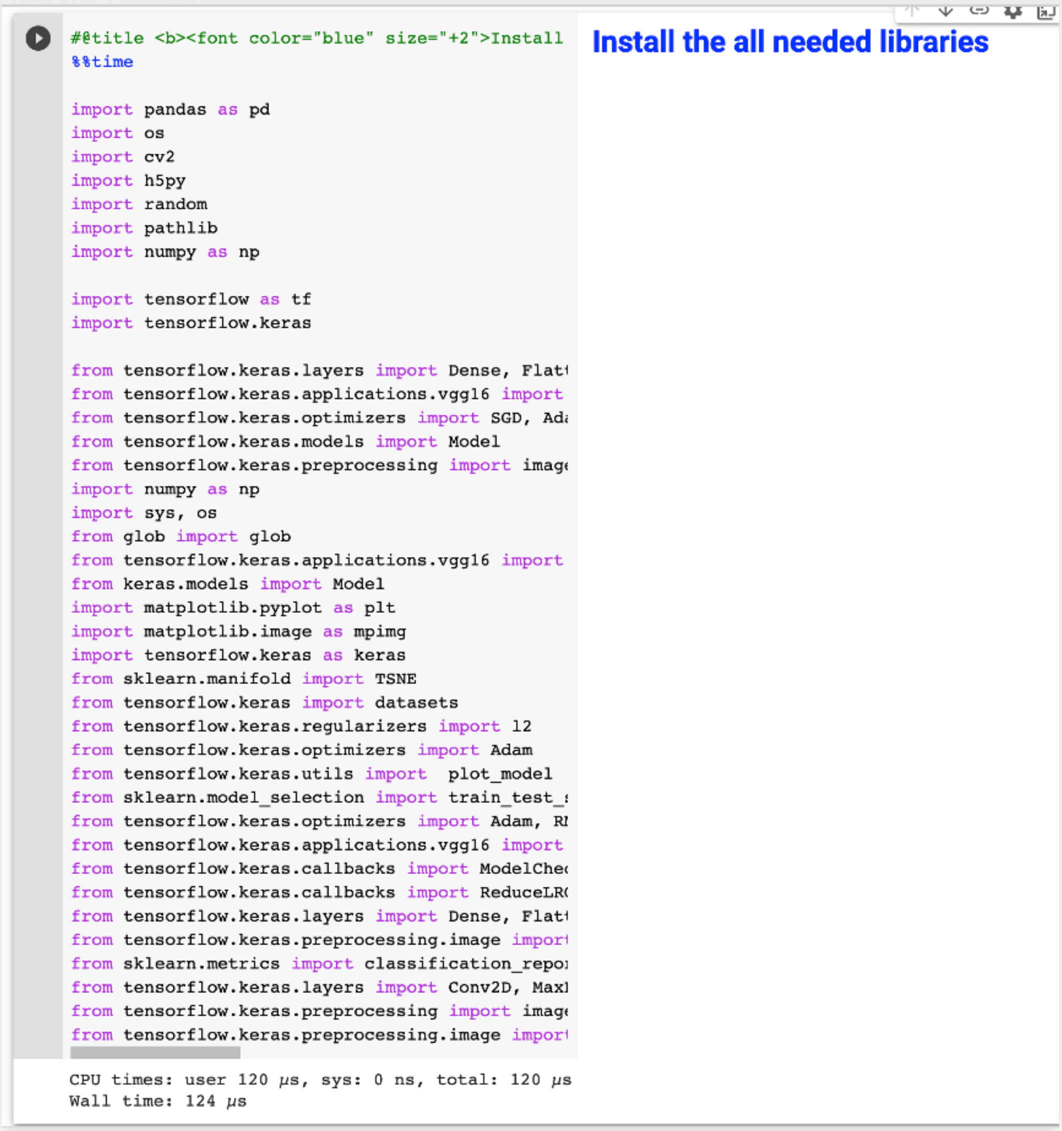


Figure 4.3: Some examples of facial images with masks in face image data set.

From two data sets, we randomly select 11600 face images, half with mask, half without mask. All face images are divided into three packages, training package, validation package and test package. The number of each package is 10000, 800 and 800, respectively. In each package, it was partitioned into sub-packages with masks and without masks.

4.2 Experimental Setup

Multiple platforms were used in the whole experiment. The model was divided into two parts, the VGG part was run on colab and kaggle website. The programming language used is python3, and the FR-TSVM part was run on Matlab(R2021a). Both parts are in a MacBook Pro (15 inches, 2019) with an Inter Core i9 processor 2.3GHz with 16 GB RAM. First, all necessary libraries were installed, such as importing deep learning frameworks TensorFlow, NumPy, random, Matplotlib and so on, as shown in Fig. 4.4. Then, the data were partitioned into training, validation and testing packages. Therefore, according to the corresponding file package, three access paths, were established to introduce the face images into the VGG model. One example of the face images path is shown in Fig. 4.5.

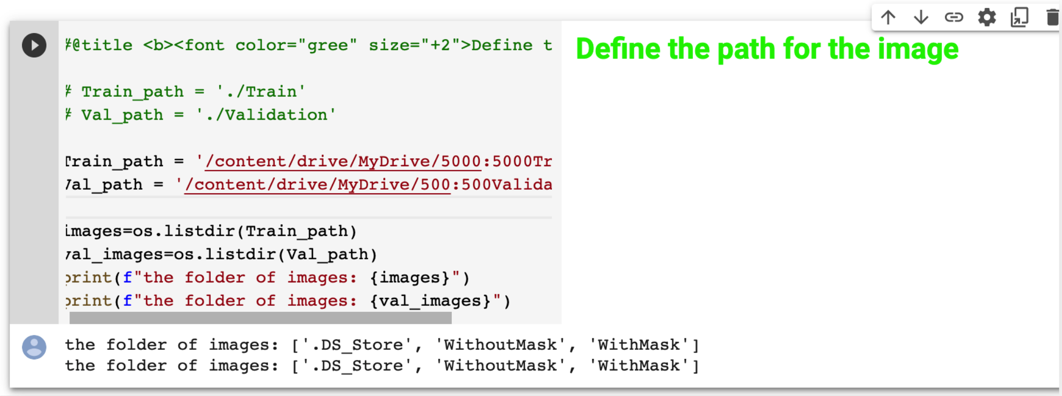


```
#@title <b><font color="blue" size="+2">Install
%%time

import pandas as pd
import os
import cv2
import h5py
import random
import pathlib
import numpy as np

import tensorflow as tf
import tensorflow.keras

from tensorflow.keras.layers import Dense, Flat
from tensorflow.keras.applications.vgg16 import
from tensorflow.keras.optimizers import SGD, Ad
from tensorflow.keras.models import Model
from tensorflow.keras.preprocessing import image
import numpy as np
import sys, os
from glob import glob
from tensorflow.keras.applications.vgg16 import
from keras.models import Model
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import tensorflow.keras as keras
from sklearn.manifold import TSNE
from tensorflow.keras import datasets
from tensorflow.keras.regularizers import l2
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.utils import  plot_model
from sklearn.model_selection import train_test_
from tensorflow.keras.optimizers import Adam, R
from tensorflow.keras.applications.vgg16 import
from tensorflow.keras.callbacks import ModelChe
from tensorflow.keras.callbacks import ReduceLR
from tensorflow.keras.layers import Dense, Flat
from tensorflow.keras.preprocessing.image impor
from sklearn.metrics import classification_repo
from tensorflow.keras.layers import Conv2D, Max
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image impor
```

Install the all needed libraries

```
CPU times: user 120 µs, sys: 0 ns, total: 120 µs
Wall time: 124 µs
```

Figure 4.4: Install the all needed libraries.



```
#@title <b><font color="gree" size="+2">Define t

# Train_path = './Train'
# Val_path = './Validation'

Train_path = '/content/drive/MyDrive/5000:5000Tr
Val_path = '/content/drive/MyDrive/500:500Valida

images=os.listdir(Train_path)
val_images=os.listdir(Val_path)
print(f"the folder of images: {images}")
print(f"the folder of images: {val_images}")
```

Define the path for the image

```
the folder of images: ['.DS_Store', 'WithoutMask', 'WithMask']
the folder of images: ['.DS_Store', 'WithoutMask', 'WithMask']
```

Figure 4.5: Define the path for the face image.

To gain the processing speedup to training the data set of 11,600 face images, a GPU was used to process a large amount of face images data firstly on colab. At the same time, the same experiment was also run on kaggle website to get the required feature vector. The run time using TPU v3-8 was compared with the running time of same code part on colab.

Input Size

In real life, all industrial display systems synthesize colors through RGB (red, green, blue), and the computer face image data storage also stores RGB three-channel pixel images. So, each face image is a three-channel pixel value matrix, and the size of the matrix is the resolution of the image. For example, a face image with a resolution of 32*32 is stored as a third-order array of 32*32*3. This is the process of one face image input into VGG model at the beginning.

In fact, the full connection layer in VGG16 is the key factor that restricts the size of input face images. Because the weight dimension of the full connection layers in VGG6 are fixed and cannot be changed, it will lead to the fixed size of all input face images. Although the last three full connected layers of VGG16 were excluded, VGG16 still can be trained, regardless of the size of face images. Convolution and max pooling layers in VGG model does not concern the size of input face images.

They just get the feature map of the previous layer and then do convolution and max pooling output. Therefore, the size of all input face images will be set to 32*32, and the preprocessed images will be input into the VGG model.

The filter of VGG model in the convolution layer specifically refers to a weight matrix. For a 32*32 face image, a 3×3 filter is multiplied by the 3×3 matrix in the face image in turn, and the corresponding convolution output is obtained. That is the generation process of face image feature vector.

Image Augmentation

Image augmentation refers to creating new face image data from existing face image data, and expecting to improve the prediction accuracy by increasing the training face image amount. For example, in digital recognition, the numbers we encounter may be tilted or rotated, so if the training face images are rotated moderately and the training amount is increased, then the accuracy of the VGG16 model may be improved. Through the operation of ”rotation”, the quality of training face image data is improved.

In this thesis, we use image augmentation and fine-tuned to increase the robustness of the overall model. First we look at a random face image in the train folder as shown in Fig. 4.6. This randomly selected face image shows a girl wearing a mask and leaning her head against the sofa.

Then, we use the image augmentation to process this face image, as shown in Fig. 4.7. Keras uses ImageDataGenerator function to set up python generator quickly. ImageDataGenerator, as its name implies, is used to generate face image data. It is used to generate a batch of face image data. During training, this function will generate data indefinitely until it reaches the specified epoch times.

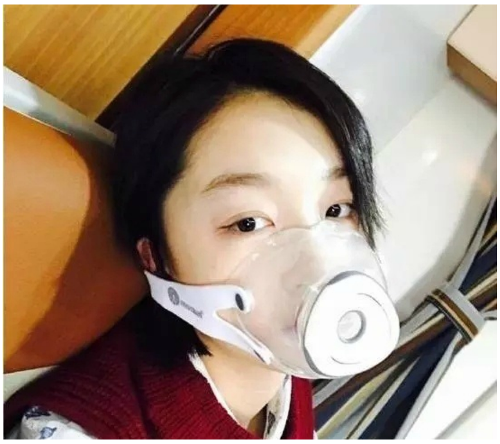
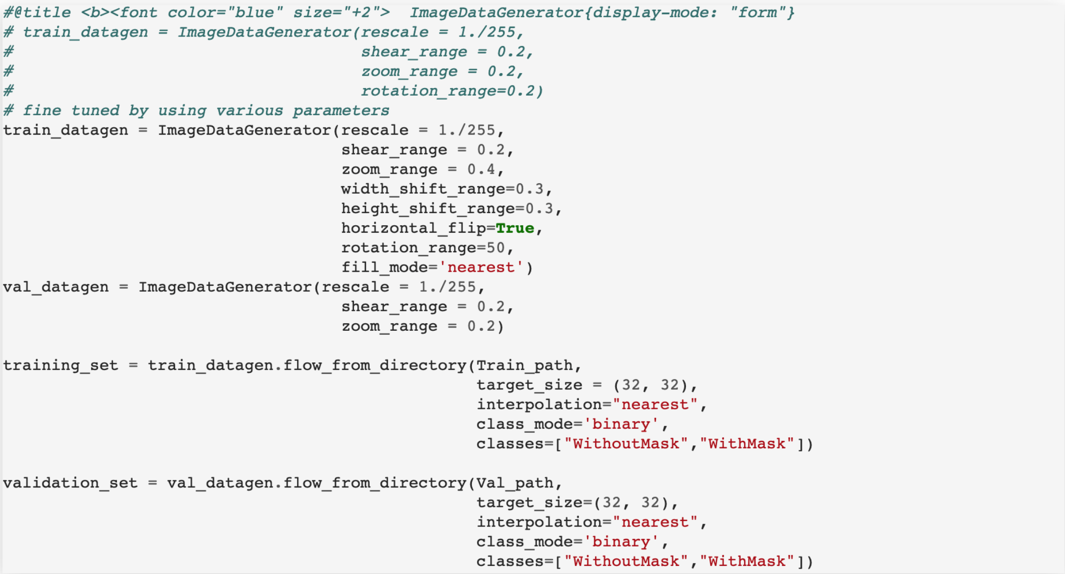


Figure 4.6: One random face image in the train folder.



```
#@title <b><font color="blue" size="+2">  ImageDataGenerator{display-mode: "form"}
# train_datagen = ImageDataGenerator(rescale = 1./255,
#                                     shear_range = 0.2,
#                                     zoom_range = 0.2,
#                                     rotation_range=0.2)
# fine tuned by using various parameters
train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.4,
                                   width_shift_range=0.3,
                                   height_shift_range=0.3,
                                   horizontal_flip=True,
                                   rotation_range=50,
                                   fill_mode='nearest')
val_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2)

training_set = train_datagen.flow_from_directory(Train_path,
                                                 target_size = (32, 32),
                                                 interpolation="nearest",
                                                 class_mode='binary',
                                                 classes=["WithoutMask","WithMask"])

validation_set = val_datagen.flow_from_directory(Val_path,
                                                 target_size=(32, 32),
                                                 interpolation="nearest",
                                                 class_mode='binary',
                                                 classes=["WithoutMask","WithMask"])
```

Figure 4.7: Image processing code.

An example of fine-tuning in the VGG model is shown in Fig. 4.8. By adjusting the original parameters in the face image, such as rotation, scaling and converting to a certain angle, the new face images are generated into our training face image dataset. This image shows a woman wearing an N95 mask. Through fine-tuning, we obtained different face images with facial postures. In the training data set, we added many different face images by fine-tuning.

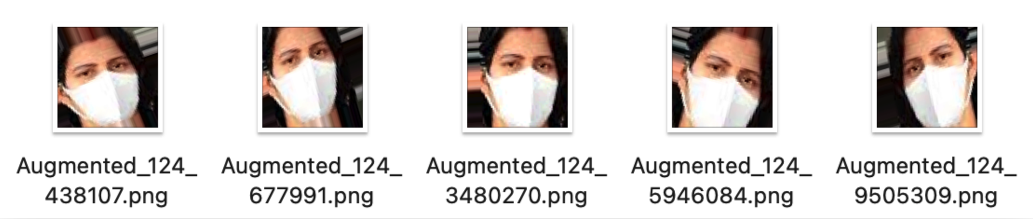
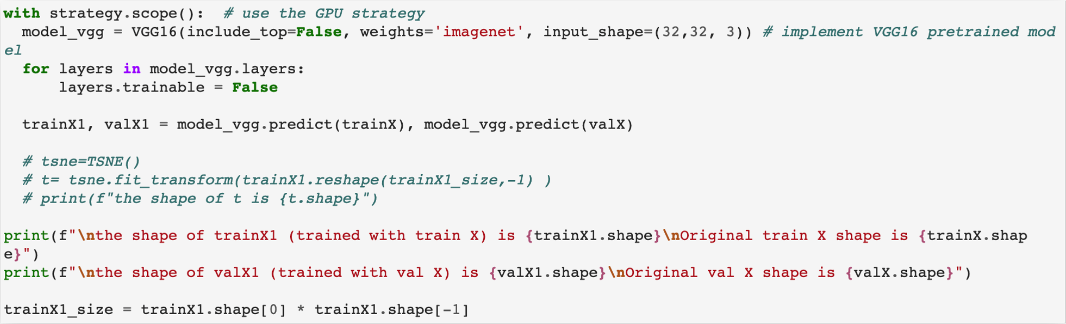


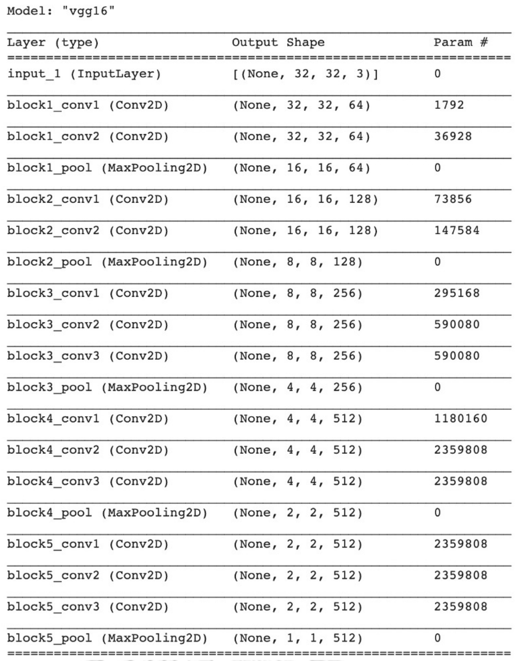
Figure 4.8: An example of a fine-tuned image. A woman wears an N95 mask.

The VGG output code is shown in Fig. 4.9. The pixel of the face image changes during the computational steps of the VGG model. For the image size of 32*32, after passing through the VGG, the size of face image pixel becomes 1*1*512. The change of specific pixels in each layer of the picture in VGG model is shown in Fig. 4.10.



```
with strategy.scope():  # use the GPU strategy
  model_vgg = VGG16(include_top=False, weights='imagenet', input_shape=(32,32, 3)) # implement VGG16 pretrained mod
el
  for layers in model_vgg.layers:
      layers.trainable = False

  trainX1, valX1 = model_vgg.predict(trainX), model_vgg.predict(valX)

  # tsne=TSNE()
  # t= tsne.fit_transform(trainX1.reshape(trainX1_size,-1) )
  # print(f"the shape of t is {t.shape}")

print(f"\nthe shape of trainX1 (trained with train X) is {trainX1.shape}\nOriginal train X shape is {trainX.shap
e}")
print(f"\nthe shape of valX1 (trained with val X) is {valX1.shape}\nOriginal val X shape is {valX.shape}")

trainX1_size = trainX1.shape[0] * trainX1.shape[-1]
```

Figure 4.9: Training images by VGG model.

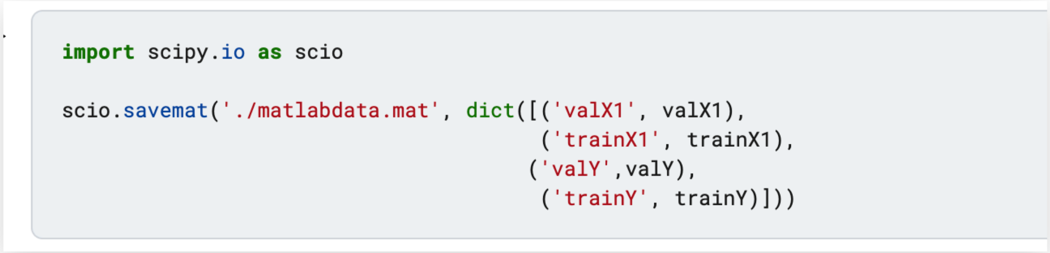


```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
input_1 (InputLayer)         [(None, 32, 32, 3)]       0         
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 32, 32, 64)        1792      
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 32, 32, 64)        36928     
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 16, 16, 64)        0         
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 16, 16, 128)       73856     
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 16, 16, 128)       147584    
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 8, 8, 128)         0         
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 8, 8, 256)         295168    
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 8, 8, 256)         590080    
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 8, 8, 256)         590080    
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 4, 4, 256)         0         
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 4, 4, 512)         1180160   
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 4, 4, 512)         2359808   
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 4, 4, 512)         2359808   
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 2, 2, 512)         0         
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 2, 2, 512)         2359808   
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 2, 2, 512)         2359808   
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 2, 2, 512)         2359808   
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 1, 1, 512)         0         
=================================================================
```

Figure 4.10: The change of specific image pixels in each layer of the VGG model.

VGG Output

A toolkit of python named scipy was adopted. It is used to save the feature vectors of face images with mask and without mask in the last max pooling layer of VGG as mat file. Each generated feature vector is used as the input of FR-TSVM model.



```
import scipy.io as scio

scio.savemat('./matlabdata.mat', dict([('valX1', valX1),
                                       ('trainX1', trainX1),
                                      ('valY',valY),
                                       ('trainY', trainY)]))
```

Figure 4.11: The code for extracting features.

One example for extracting features from VGG model is shown in Fig. 4.11.

4.3 FR-TSVM Part

The extracted image features were classified by FR-TSVM and used as the final output. In this experiment, we used a linear FR-TSVM model, and the related parameters were set as follows: CC is 100, CR is 60, V is 10. The cd algorithm was used in FR-TSVM model. One example of using FR-TSVM model to get result is shown in Fig. 4.12.

In FR-TSVM result, -1 means with mask and 1 means without mask. The results were evaluated by accuracy, precision and recall. The tested data set has 800 face images, including 400 face images with masks and 400 face images without masks.The values of outclass and valY are shown in Fig. 4.13 and Fig. 4.14, respectively. Outclass shows the correct face images, with masks labeled -1 and without masks labeled 1. valY shows the results after using the whole model, with masks labeled -1 and without masks labeled 1. According to the test results, the face mask samples can be divided into four categories, and the following are the specific explanations of these four categories.

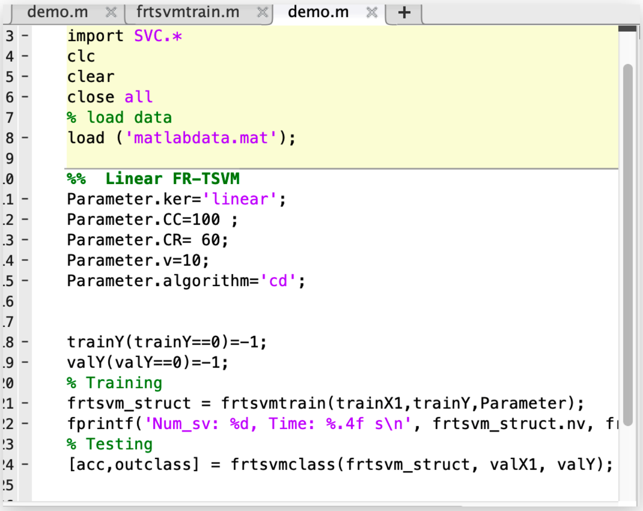
True Positive: The true category of the face mask dataset is a positive example, and the result predicted by the model, which is correct.

True Negative: the true category of the face mask dataset is negative, and the model predicts it as negative, and the prediction is correct.

False Positive: The true category of the face mask dataset is a negative case, but the model predicts it as a positive case and the prediction is wrong.

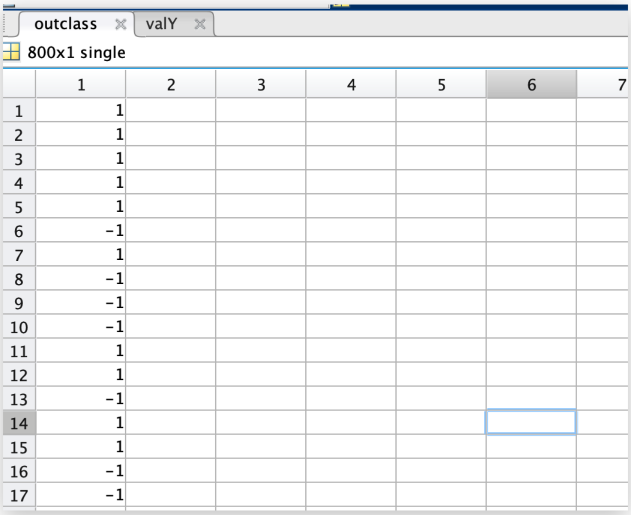
False Negative: The true category of the face mask dataset is a positive example and the prediction is wrong.

Through the comparison between outclass and valY, we can get the relevant indicators, such as TP/FP/TN/FP in the face image data set.



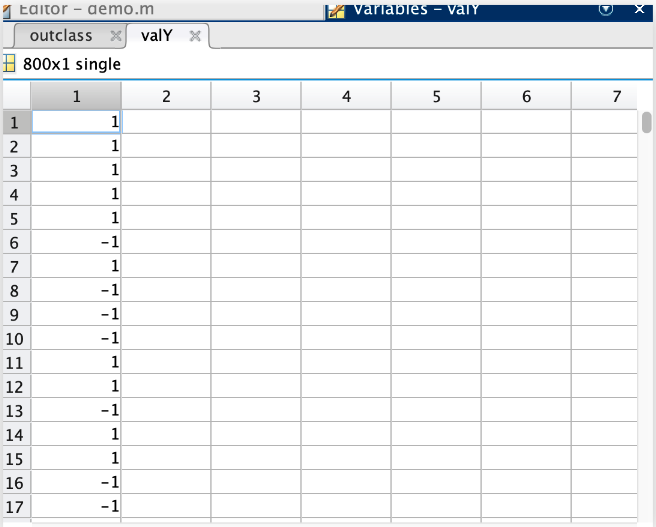
```
import SVC.*
clc
clear
close all
% load data
load ('matlabdata.mat');

%%  Linear FR-TSVM
Parameter.ker='linear';
Parameter.CC=100 ;
Parameter.CR= 60;
Parameter.v=10;
Parameter.algorithm='cd';


trainY(trainY==0)=-1;
valY(valY==0)=-1;
% Training
frtsvm_struct = frtsvmtrain(trainX1,trainY,Parameter);
fprintf('Num_sv: %d, Time: %.4f s\n', frtsvm_struct.nv, f
% Testing
[acc,outclass] = frtsvmclass(frtsvm_struct, valX1, valY);
```

Figure 4.12: Using FR-TSVM model to get result.



	1	2	3	4	5	6	7
1	1						
2	1						
3	1						
4	1						
5	1						
6	-1						
7	1						
8	-1						
9	-1						
10	-1						
11	1						
12	1						
13	-1						
14	1						
15	1						
16	-1						
17	-1						

Figure 4.13: Outclass in FR-TSVM model.



	1	2	3	4	5	6	7
1	1						
2	1						
3	1						
4	1						
5	1						
6	-1						
7	1						
8	-1						
9	-1						
10	-1						
11	1						
12	1						
13	-1						
14	1						
15	1						
16	-1						
17	-1						

Figure 4.14: ValY in FR-TSVM model

4.4 Output of Combined Model

After the face image passes through the first 13 layers of the pre-VGG model, the train loss is 0.7376 and the accuracy is 0.9355, val_loss is 0.6044, and the val_accuracy is 0.9399 as shown in Table 4.1. The related loss is shown in Fig. 4.15, and the related accuracy is shown in Fig. 4.16.

Table 4.1: THE TRAIN LOSS AND VAL LOSS

10*Epoch	Train loss	Train accuracy	val loss	val accuracy
VGG 1/10	19.4477	0.3740	9.2656	0.5085
VGG 2/10	7.2103	0.5537	2.7545	0.7528
VGG 3/10	2.4571	0.7794	1.5417	0.8619
VGG 4/10	1.4316	0.8813	1.1703	0.9609
VGG 5/10	1.2224	0.9051	1.0679	0.9089
VGG 6/10	1.0672	0.9177	0.8405	0.9169
VGG 7/10	0.8546	0.9233	0.9131	0.9289
VGG 8/10	0.7273	0.9371	0.8353	0.9219
VGG 9/10	0.6544	0.9375	0.6751	0.9419
VGG 10/10	0.7376	0.9355	0.6044	0.9399

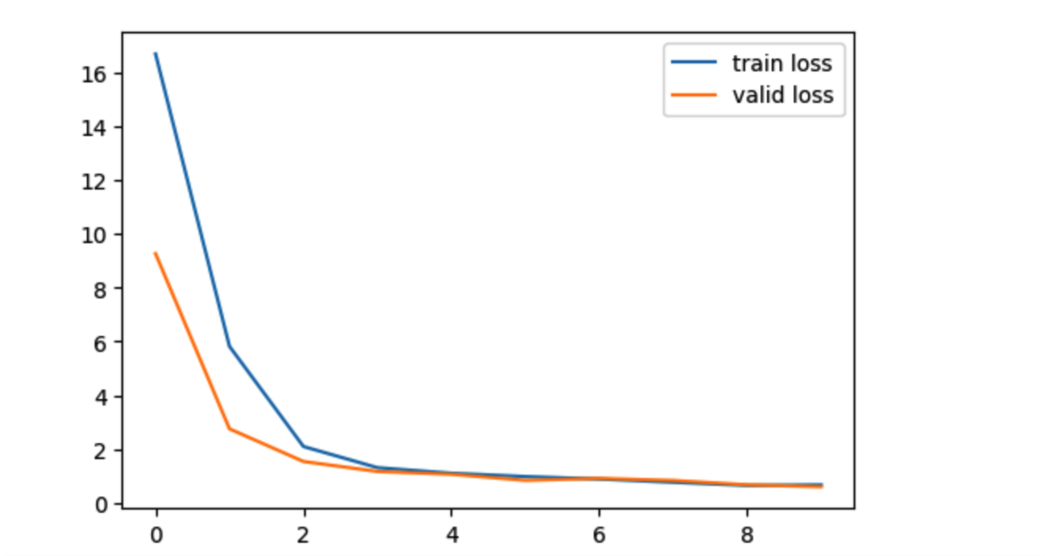


Figure 4.15: Loss-curve in this model.

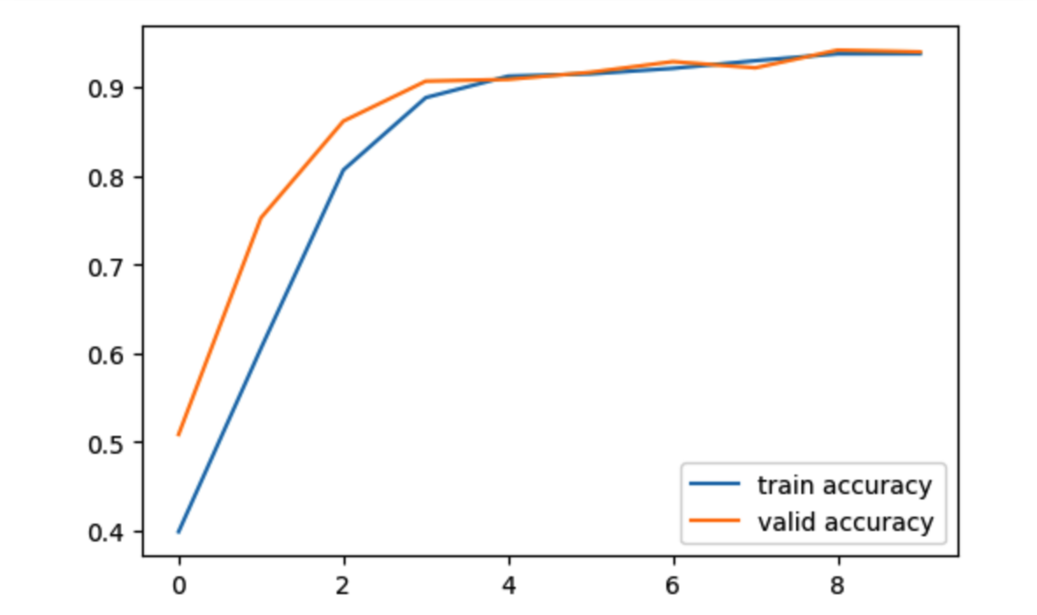


Figure 4.16: The accuracy curve in this model.

It is difficult to adjust the parameters of FR-TSVM in the experiment. After properly changing the parameters, the final result has not changed much. We used the same 10,000 face mask images from WIDER and MAFA data sets to train and the accuracy is 95.125%. The result by the proposed method as shown in Table 4.2. The accuracy of wearing masks is 94.5%, precision is 95.88%,recall is 93%, F1 is 94.42%; The accuracy of not wearing masks is 95.75%, precision is 96.45%, recall is 95%, F1 is 95.72%.

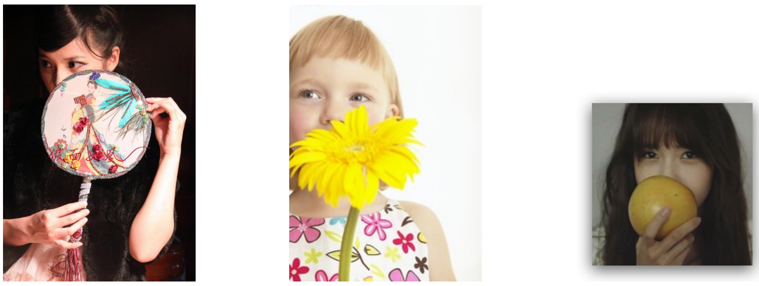
Table 4.2: RESULTS FROM FR-TSVM

4*Results	Accuracy	Precision	Recall	F1
Detect images	95.125%	95.43%	94.71%	95.07%
Detect images with masks	94.5%	95.88%	93.0%	94.42%
Detect images with masks	95.75%	96.45%	95.0%	95.72%

Difficult Images

In a large number of face mask images. There are some difficult images to detect by this proposed method, such as Fig. 4.17. In pic1, a beautiful girl blocked her half face with a fan. In pic2, a lovely little girl covered her face with sunflowers. And in pic3, a beautiful girl covered her mouth with a grapefruit.

From the above examples, we can find some rules. When the mask part of face images is replaced by fans, sunflowers, fruits and other objects, these kinds of face images still cannot be correctly identified by our team's model.



(a) pic1. (b) pic2. (c) pic3.

Figure 4.17: Difficult images to detect by proposed method.

4.5 Comparative Other Methods

To verify the performance of the models used by our team, we compared the models with YOLO, M-CNN and VGG models. Face images from WIDER and MAFA are used in the data sets, with 10,000 face images used for training, 800 face images for validation and 800 face images for test. Our model was compared by the test results of accuracy, precision, recall and confusion matrix with other three models.

YOLO

The results were reported in box, objectness, classification, mAP, precision and recall. *Box* means that GIOU Loss is used as the loss of bounding box. Box is presumed to be the mean value of GIOU loss function, and the smaller the box, the more accurate it is. *Objectness* means that it is presumed to be the average loss of target detection, and the smaller the target face image detection, the more accurate it is. *Classification* means that it is presumed to be the average value of classification loss, and the smaller the classification, the more accurate it is. *mAP* is mean Average Precision. mAP refers to the area enclosed after drawing with Precision and Recall as two axes, m represents the average, and the number after represents the threshold for judging as positive and negative samples. For example, mAP@0.5: indicates the average mAP with a threshold greater than 0.5.

By using YOLOv3 algorithm, the accuracy of YOLOv3 is 85.02%, precision is 69.02%, recall is 87.74%. The result is shown in Fig. 4.18. The precision-recall curve is shown in Fig. 4.19.

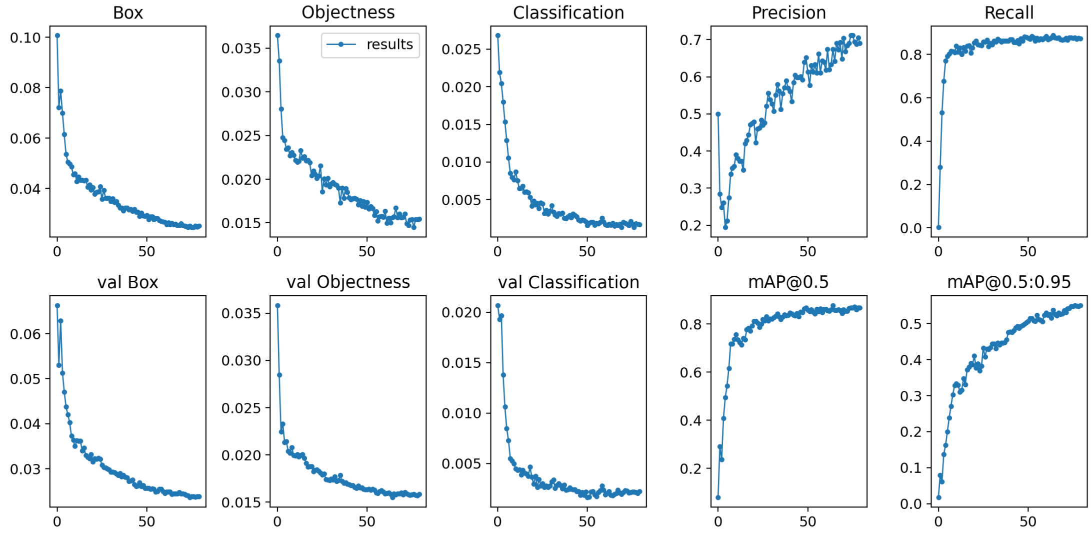


Figure 4.18: The results from using YOLOv3.

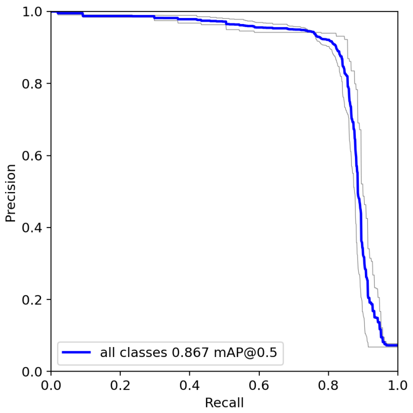


Figure 4.19: The results from using YOLOv3 based on parameters.

Mask-CNN

We already know that CNN is used for image recognition. But if we want to identify the same species, for example, the kind of mask one person wears, it is difficult to identify with CNN. At this time, we consider using Mask-CNN model to identify some key parts of objects. The whole design of M-CNN is based on full convolution net. The second way can be used to focus on the nose, mouth and chin of the face to identify whether someone wears a mask in the face picture. M-

CNN was downloaded from GitHub. The parameters were pre-trained on ImageNet classification.

The accuracy by use M-CNN is 91.57%, precision is 88.6%, recall is 94.15%.

VGG

The third method, we considered using VGG model for face image classification. In the third VGG experiment, 11600 face images will be selected from WIDER and MAFA as experimental data sets. By training the VGG16 model, we can judge whether a face wears a mask or not. The relevant parameters of the model were set on imagenet at first. Because there are a lot of parameters to be calculated in the last three fully connected layers, this will greatly increase the parameters we need to train. The obvious disadvantage of this VGG model is that it may be over-fitted and slow down the training speed of the model

The accuracy by use only VGG16 for face image classification is 93.2%, precision is 91.6%, recall is 94.5%.

Result

Table 4.3: COMPARATIVE RESULTS

4*Method	Accuracy	Precision	Recall
Li, C., et al. [13]	85.02%	69.02%	87.74%
Rao, T.S., et al. [14]	91.57%	88.6%	94.15%
VGG16	93.2%	91.6%	94.5%
Proposed method	95.125%	95.43%	94.71%

From Table 4.3, the precision of the first YOLO is obviously lower than 70%, and the overall accuracy is lower than the other three methods. The third VGG method seems to be superior to the first and second methods, with an accuracy of 93.2%, precision of 91.6% and recall of 94.5%, which is 8.18% higher than the first method and 1.63% higher than the second method. However, the accuracy of our method is 1.925% higher than the third method, and precision is significantly higher than VGG, which is 3.83% higher. By comparing the above results, we can find use this proposed method to detect people wear masks or not is excellent in accuracy, precision and recall, which proves that the model has good performance.

Chapter V

ANALYSIS

To judge the quality of this proposed model, we should look at the indexes of accuracy, precision, recall and so on. The previous experimental comparison results have proved that this proposed method has higher accuracy, precision and recall than the other three methods.

The confusion matrices of YOLOv3, M-CNN, VGG and this experiment were compared respectively in the same environment, and the test set used the same face images from WIDER and MAFA datasets, with a total of 800 face images, including 400 face images with masks and 400 face images without masks.

Confusion matrix concept here is a summary of the prediction results of face image classification problems. It is the key to confuse the matrix that the number of correct and incorrect predictions is summarized by the count value and subdivided by each class. The confusion matrix shows which part of face image classification model will be confused when forecasting. It not only lets us know the mistakes made by the face image classification model, but also gives more importantly, what types of mistakes are happening. It is this decomposition of results that overcomes the limitation of using only classification accuracy.

The results of face image classification can be seen from the confusion matrix. In the same test set, the correct number of samples measured by the proposed model is 385(TP)+376(TN), and a total of 761 face images are accurately detected in 800 test images. It can be seen that this experimental model has an excellent classification effect on whether a face wears a mask or not.

From Table 5.1, the TP, TN applicable to proposed model are 385, 376 respectively. Compared with the first and second methods, this proposed method can

Table 5.1: COMPARISON OF CONFUSION MATRIX

4*Methods	TP	FP	FN	TN
Li, C., et al. [13]	337	63	57	343
Rao, T.S., et al. [14]	376	24	43	357
VGG16	383	17	37	363
Proposed method	385	15	24	376

recognize more TP face images, and the number of face images for TN recognition is more than the other three methods. The FP and FN numbers of this proposed method are obviously smaller than those of the first and second methods, which reflects that the accuracy of this proposed method is higher than the first and second methods; The FN number of this proposed method differs from the third method by 13 face images. It can also be seen that the proposed method in this paper can identify more images without masks, and it is superior to the third method in the accuracy of detecting without masks.

Table 5.2: COMPARATIVE OF PROS AND CONS

4*Method	Pros	Cons
Li, C., et al. [13]	Automatically	Low precision
Rao, T.S., et al. [14]	Fewer parameters	Low precision
VGG16	Model stability	Slow training
Proposed method	High accuracy and fast	Identify occluded pictures

From the Table 5.2, we can compare the main pros and cons of the four methods. The first method uses YOLOv3 to detect whether a face image wears a mask or not. The YOLO v3model has been trained in advance. During the detection process, the system can automatically detect whether a face picture wears a mask. However, according to the experimental results, the precision of using YOLO is very low, only 69.02%, and the total accuracy rate is only 85.02%.

The second method is the optimization of CNN model. In M-CNN model, Adam as the optimizer, and cross entropy as the loss function network. The model is simple in structure and has the advantage that few parameters are used in the calculation. The precision obtained by using M-CNN model is low, compared VGG and proposed method only 88.6%.

The third one is the VGG16 model, which is stable and has a multi-layer convolution network. It is a good choice for training face images, but there are a lot of parameters to be calculated in the last three full connection layers of VGG16, which leads to a long time for training face images.

The proposed method is to remove the last three layers of VGG16, and then use FR-TSVM to classify human face images. Compared with the first three methods, the experimental results show that the proposed model has better performance in detecting human face images. After VGG16 removes the last three layers of fully connected layers, the calculation amount of parameters decreases, the occupied memory decreases, and the overall proposed model calculation speed of the experiment is extremely fast. But the biggest defect is that the proposed method cannot effectively identify some occluded face images.

Chapter VI

DISCUSSION AND CONCLUSION

6.1 Summary of Findings

In this paper, we use a combination of VGG and linear FR-TSVM to detect whether people wear masks or not. The number of samples is 11600 images, we find that the learning speed of the proposed model is fast, the memory occupied is small, and various types of masks can be detected with high accuracy of 95.125%.

Because a large number of face images of different scenes, masks, ages, genders, and differences in dress, were trained at the beginning, and image augmentation was used in the VGG model, this proposed model is also suitable for different scenes and masks, and can detect different face images with different face postures.

However, this proposed model is not applicable to face images with various types of occlusion. For example, people who take quilts to drink water or eat fruit can not be accurately identified by the proposed model. Theoretically, the nonlinear FR-TSVM model can obtain better accuracy, but it is difficult to adjust the parameters of the nonlinear FR-TSVM model to obtain better experimental results, which depends on the selection of data sets.

Another limitation of this proposed model is that it uses static face images. The images selected in this paper are in JPG format and are static face images. In real life, it is necessary to combine this proposed model with the latest AI technology. For example, if this method is used to detect whether the driver is wearing a mask during work time, it is necessary to consider replacing the static face image with the dynamic face image, which can effectively detect whether the driver is wearing a mask in the video. In this way, it can be detected whether drivers wear masks or not at different time periods, and only static images can not ensure that drivers always wear masks during work.

6.2 Future Work

In this paper, a deep transfer learning model is used, which combines VGG and FR-TSVM model, in which VGG is used as the feature extractor of face images, while FR-TSVM is used as the classification of whether face images wear masks or not. The model in this paper can be optimized from different angles.

From VGG part, the improved method of VGG can be to compare the differences in validation sets using different network corrections, such as different tuning methods, all-layer tuning or all-connected-layer tuning. We can also improve the VGG model in other ways. For example, choosing a layer of global max pooling to replace the full connection layers of VGG16 model can greatly reduce the parameters we need to train. This is of great help to simplify the face image experimental model and improve the training speed of the VGG model.

The proposed model still has a certain distance from real-time detection of whether people wear masks or not. One reason is that the selected face images are static images. In the future, we can consider using dynamic face images or videos as the input of VGG model, and we can use VGG model to successfully obtain face image features after training a large number of face image samples.

From FR-TSVM part, theoretically, the performance of nonlinear FR-TSVM is better than that of linear FR-TSVM, but the parameters of nonlinear FR-TSVM are difficult to adjust to obtain the highest accuracy. This requires a large number of face image samples to be tested and manually adjusted again and again in the future. Moreover, according to the different face samples in the data set, the parameters for obtaining the best performance of the nonlinear FR-TSVM model will also change.

The experimental results show that this proposed model can’t correctly identify partially occluded face images. This problem can be solved by optimizing the model. Or, the proposed model can be combined with PSC-Net method to improve the accuracy of recognition of occluded face images and optimize the performance of the model.

REFERENCES

[1] Tang, B., et al., An updated estimation of the risk of transmission of the novel coronavirus (2019-nCov). Infectious Disease Modeling, 2020. 5: p. 248-255.

[2] Tang, B., et al., Estimation of the Transmission Risk of the 2019-nCoV and Its Implication for Public Health Interventions. Journal of Clinical Medicine, 2020. 9(2): p. 462.

[3] World Health Organization, Mask use in the context of COVID-19, Interim guidance 1 December 2020.

[4] Jeremy Howard, et al, An evidence review of face masks against COVID-19, PNAS 2021 Vol. 118 No. 4 e2014564118.

[5] Jayadeva, R. Khemchandani, and S. Chandra, Twin Support Vector Machines for Pattern Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2007. 29(5): p. 905-910.

[6] Peng, X., A ν-twin support vector machine (ν-TSVM) classifier and its geometric algorithms. Inf. Sci., 2010. 180: p. 3863-3875.

[7] Gao, B.-B. and J. Wang, A Fast and Robust TSVM for Pattern Classification. ArXiv, 2017. abs/1711.05406.

[8] Li, Q., et al., Early Transmission Dynamics in Wuhan, China, of Novel Corona virus–Infected Pneumonia. New England Journal of Medicine, 2020. 382(13): p. 1199-1207.

[9] Chan, J.F.-W., et al., A familial cluster of pneumonia associated with the 2019 novel corona virus indicating person-to-person transmission: a study of a family cluster. The Lancet, 2020. 395(10223): p. 514-523.

[10] Zhou, T., et al., Preliminary prediction of the basic reproduction number of the Wuhan novel corona virus 2019-nCoV. Journal of Evidence-Based Medicine, 2020. 13(1): p. 3-7.

[11] Read, J.M., et al., Novel corona virus 2019-nCoV: early estimation of epidemiological parameters and epidemic predictions. 2020, Cold Spring Harbor Laboratory.

[12] Inamdar, M. and N. Mehendale, Real-Time Face Mask Identification Using Facemasknet Deep Learning Network. SSRN Electronic Journal, 2020.

[13] Li, C., J. Cao, and X. Zhang, Robust Deep Learning Method to Detect Face Masks, in Proceedings of the 2nd International Conference on Artificial Intelligence and Advanced Manufacture. 2020, Association for Computing Machinery: Manchester, United Kingdom. p. 74–77.

[14] Rao, T.S., et al., A Novel Approach To Detect Face Mask To Control Covid Using Deep Learning. European Journal of Molecular & Clinical Medicine, 2020. 7(6): p. 658-668.

[15] Worby, C.J. and H.-H. Chang, Face mask use in the general population and optimal resource allocation during the COVID-19 pandemic. Nature Communications, 2020. 11(1).

[16] Chowdary, G.J., et al. Face Mask Detection using Transfer Learning of Inception V3. in BDA. 2020.

[17] Loey, M., et al., A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement, 2021. 167: p. 108288.

[18] Wang, W., et al., High-Resolution Radar Target Recognition via Inception-Based VGG (IVGG) Networks. Computational Intelligence and Neuroscience, 2020. 2020: p. 8893419.

[19] Wei, J., et al., Analyzing the impact of soft errors in VGG networks implemented on GPUs. Microelectronics Reliability, 2020. 110: p. 113648.

[20] Xu, X., et al., Perceptual-Aware Sketch Simplification Based on Integrated VGG Layers. IEEE Transactions on Visualization and Computer Graphics, 2021. 27(1): p. 178-189.

[21] Zhang, D., J. Lv, and Z. Cheng, An Approach Focusing on the Convolutional Layer Characteristics of the VGG Network for Vehicle Tracking. IEEE Access, 2020. 8: p. 112827-112839.

[22] Zhou, Y., et al., Improving the Performance of VGG Through Different Granularity Feature Combinations. IEEE Access, 2021. 9: p. 26208-26220.

[23] Batagelj, B., et al., How to Correctly Detect Face-Masks for COVID-19 from Visual Information? Applied Sciences, 2021. 11(5): p. 2070.

[24] Verberne, J.W.R., P.R. Worsley, and D.L. Bader, A 3D registration methodology to evaluate the goodness of fit at the individual-respiratory mask interface. Computer Methods in Biomechanics and Biomedical Engineering, 2020: p. 1-12.

[25] A. Messac, Optimization in Practice with MATLAB!R : For Engineering Students and Professionals, Cambridge University Press, 2015.

Appendix I

SAMPLE CHAPTER

A.1 Features

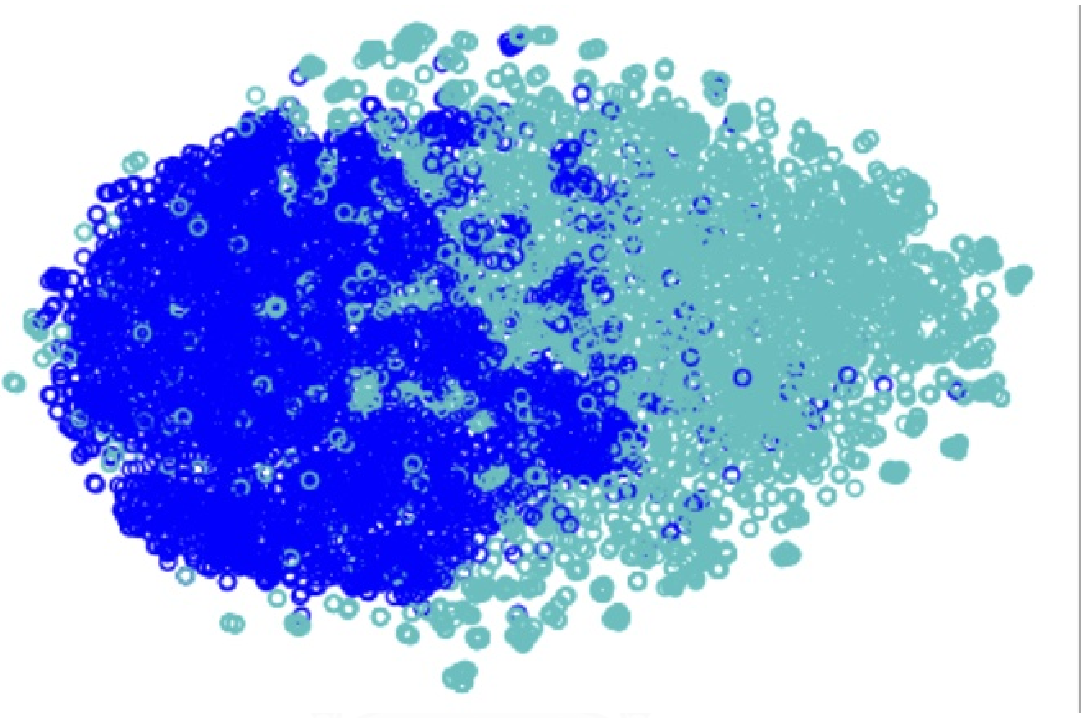


Figure A.1: Features extracted from the VGG model

As shown in the Fig. A.1, this paper uses TSNE tool to visualize the masked vector and unmasked vector extracted from VGG16 model, in which green represents the masked feature vector and blue represents the unmasked feature vector. From the Fig. A.1, the two vectors are evenly distributed, and it is feasible to use linear FR-TSVM to find the hyperplane between two feature vectors. It can also be seen that the result of extracting face image feature vectors using VGG16 is very good.

Appendix II

LIST OF PUBLICATIONS

B.1 International Conference Proceeding

1. HangKai Wang, & Chidchanok Lursinsap. Detecting Facial Images In Public With And Without Masks Using VGG And FR-TSVM Models. In 2021 18th International Joint Conference on Computer Science and Software Engineering (JCSSE).

Biography

My name is Hangkai Wang. I was born on October 9th, 1995. My home address is in Busen Avenue, Fengqiao, Zhuji, Shaoxing, Zhejiang Province, China. I studied at Wenzhou Medical University as an undergraduate, and then went to Chulalongkorn University, where I studied computer science and meet a lot of wonderful persons. I love my happy time in Chulalongkorn University. Here, many teachers and classmates are kind and willing to give me great help during my school time. Thank you all.