User Guide

1. Introduction

1.1 Background

Nowadays, COVID 19 threatens the safety of each of us. A very important preventive measure is to wear a mask. Although most people wear masks in public facilities, there are still a few who break the rules. In order to help managers find these people who do not wear masks and remind them through some methods, we developed this "Wear a Mask" system.

1.2 Aim of Design

Our system aims to provide a solution to deal with rule-breakers. The system can be applied to 7-11 convenience stores, Miniso, Watsons and other facilities. Customers collect faces and information when they enter the facility. Our system uses surveillance cameras in the store for detection. When a customer removes the mask in the store, we will capture the person's face image, and use the face identification to find the person from the database collected at the entrance. When this person is found, our system will notify this person by SMS or email, reminding him to wear a mask.

Our system is designed to provide a set of fully automatic security system for some public facilities. The system is convenient for facility managers to ensure the safety of facilities.

2. Usage Instructions

2.1 Software Description:

Our software is divided into two parts: 1. Entrance Face Detection Module; 2. Image Enhancement Module; 3. In-store Mask Detection and Face Identification Module.

2.1.1 Entrance Face Detection Module:

The camera performs real-time face detection, and when a user enters information at the entry, it stores the face picture and the entered data.

2.1.2 Image Enhancement Module:

The pictures captured by the camera may have some situations that are not conducive to mask detection and face recognition, such as blurred pictures, too small pictures, noisy pictures, insufficient lighting, etc. In response to these situations, we use image enhancement technology to perform processing such as noise reduction, sharpening, histogram equalization, and super-resolution reconstruction on the image, which greatly enhances the accuracy of mask detection and face recognition.

2.1.3 In-store mask detection and face identification:

The in-store camera provides masks for trial detection. When a customer takes off the mask, the system automatically captures the person's image, and then performs face identification and finds the customer from the database. Finally, use the information entered by the customer to identify him.

2.2 Operating Environment:

2.2.1. Hardware:

1. Computer, 2. Surveillance camera.

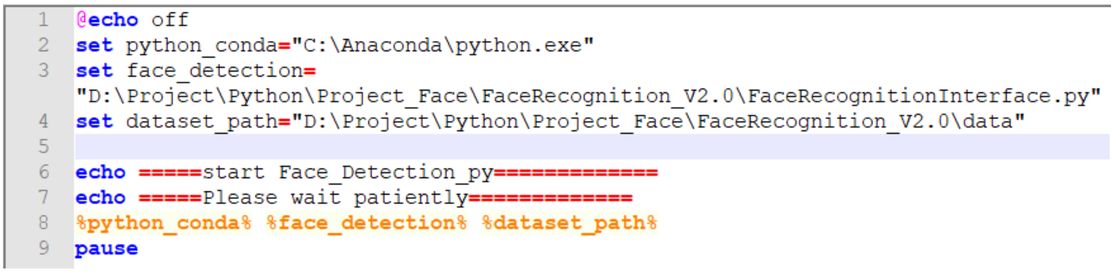
2.2.2. Support Software:

1) System: Win10;
2) Language: Anaconda python 3.7 base;
3) package:
For face detection.
    opencv-python== 4.4.0.44;
    numpy ==1.16.1;
    threading;
    tensorflow-gpu==2.3;
    keras ==2.4.3;
    cudnn==7.6.5;
    cuda==10.1;
    pillow==7.2.0
    matplotlib==3.3.2
For mask detection.
    tensorflow>=1.15.2
    keras==2.3.1
    imutils==0.5.3

matplotlib==3.2.1
scipy==1.4.1

2.3 Software Usage

2.3.1 Entrance Face Detection Module



```
@echo off
set python_conda="C:\Anaconda\python.exe"
set face_detection="D:\Project\Python\Project_Face\FaceRecognition_V2.0\FaceRecognitionInterface.py"
set dataset_path="D:\Project\Python\Project_Face\FaceRecognition_V2.0\data"

echo =====start Face_Detection_py=============
echo =====Please wait patiently=============
%python_conda% %face_detection% %dataset_path%
pause
```

1) Configure run_face_detection.bat:
    python= "python path"
    face_detection=”path of python file”
    dataset_path = "The path of the database"
2) Use CMD of Anaconda to run the run_face_detection.bat

2.3.2 Image Enhancement

The image enhancement program has been embedded in our system, and the program will be automatically called without external instructions. If you need to use image enhancement directly, you can use the image_enhance.py file, and the program will automatically enhance the image.

2.3.3 In-store Mask Detection and Face Identification Module

1) install the above dependencies
2) dataset contains with mask and without mask, you can view the dataset in the dataset folder.
3) the trained model - mask_detector model is given along the code, keep both in the same directory for easy navigation
4) In case you want to train the model again. Run :
training_model_mask_detector.ipynb
5) load the mask_detector in the file : Mask_video_Final.ipynb
6) open this ipython file , make sure you're able to initialize the video stream with your camera
7) If you have multiple cameras to work with change the index as required
8) (VideoStream(src=0)). You can change the src value if youre not using primary camera
9) now just load the model and run the code