Maskit: COVID-19 Resiliency through Computer Vision and Robotics

Allen Mao
University of California, Berkeley
Berkeley, California
allenmao@berkeley.edu

Hans Gundlach
University of California, Berkeley
Berkeley, California
11235hans@berkeley.edu

CONTENTS

I. INTRODUCTION

The COVID-19 pandemic has led to, at the time of this writing, almost half a million deaths, the worst global recession since the Great Depression, and school closures that have affected nearly all of the world's student population. Although reopening efforts have been underway in many parts of the world, their results have been mixed and it is often instead in citizens' own hands to protect themselves from this virus through means such as social distancing, frequent handwashing, and mask wearing. Although the United States Centers for Disease Control and Prevention (CDC) has issued recommendations for citizens to wear masks to reduce exposure to the virus, the choice to wear a mask has unfortunately evolved to become a political question too [1], [2]. Apart from CDC and World Health Organization (WHO) guidelines to wear facial coverings, i.e. masks, in this article we will not discuss the merits of face coverings; this knowledge is assumed. Instead, we propose, describe, and demonstrate *Maskit*, a computer vision and robotics system that keeps business owners and other ordinary citizens safe by blocking people who fail to wear a mask while letting people who do through entrances.

A. Code Source

Our work is found at https://github.com/hansgundlach/FaceMaskDetectionRasPi. It is based on earlier an earlier repository by AIZoo Tech[1]. We thank AIZoo Tech for releasing their models to the public with the MIT License. Throughout this paper, we detail our original contributions on top of AIZoo Tech's technology to enhance public safety during a worldwide pandemic using computer vision and robotics.

## II. PURPOSE & MOTIVATION

A. Meet the Team

The team: Allen to the left, Hans to the right

We are both undergraduate students at the University of California, Berkeley. Allen is a Computer Science major who is primarily interested in Natural Language Processing

---

[1]https://github.com/AIZOOTech/FaceMaskDetection

B. Previous Work on Mask Identification with Computer Vision

In the past several months, several organizations, for example Didi Chuxing[2] and AI Zoo Tech[3] have released open-source pre-trained neural network models on multiple platforms (Keras, Tensorflow, and Caffe) that classify faces within a photo or video as mask-wearing or not. Both models perform spectacularly as the Didi model was trained on a dataset of over 200,000 faces while the AIZOO model consists of 24 convolutional layers trained on approximately 8000 images from WIDER Face and MAFA datasets. Accordingly, the Didi model achieves at least 98% accuracy and the AIZoo model has a 0.896 ROC AUC (*receiver operating characteristic area under the curve* for face detection and 0.919 ROC AUC for face mask detection, as seen in their curve below.

C. Meet Maskit.

Maskit

[2]https://github.com/didi/maskdetection

[3]https://github.com/AIZOOTech/FaceMaskDetection
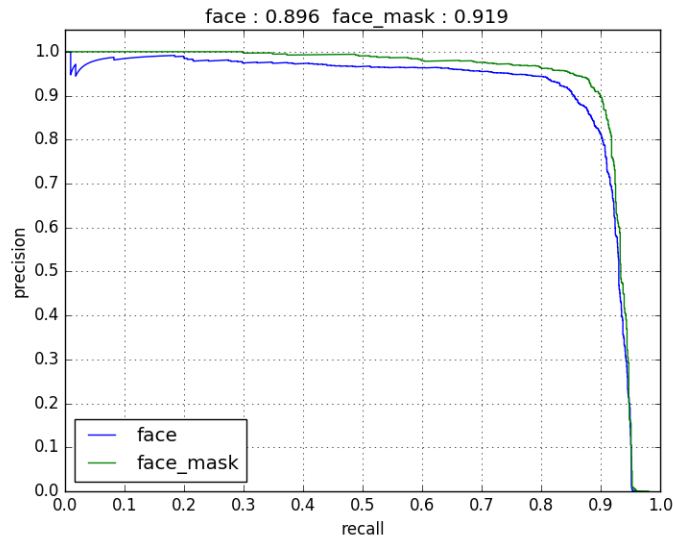
III. STEP BY STEP USAGE INSTRUCTIONS

A. Setting up the server

B. Running it on R Pi

```
# upload_img_post.py
import os
import requests
import json
import wiringpi
from io import BytesIO
from time import sleep
from picamera import PiCamera
url = 'http://192.168.0.31:8000'
wiringpi.wiringPiSetupGpio()
wiringpi.pinMode(18, wiringpi.GPIO.PWM_OUTPUT)
wiringpi.pwmSetMode(wiringpi.GPIO.PWM_MODE_MS)
wiringpi.pwmSetClock(192)
wiringpi.pwmSetRange(2000)
delay_period= .001


stream = BytesIO()
camera = PiCamera()
camera.start_preview()
sleep(2)
for i in list(range(2)):
    camera.capture(stream, 'jpeg')
    stream.seek(0)
    data = stream.read()
    r = requests.post(url,data=data)
    lst_str = r.content.decode('utf-8')
    lst = json.loads(lst_str)
    all_masked  = sum([x[0] for x in lst])
    if (all_masked == 0):
        for pulse in range(50, 250, 1):
            wiringpi.pwmWrite(18, pulse)
            sleep(delay_period)
        sleep(3)
        for pulse in range(250, 50, -1):
            wiringpi.pwmWrite(18, pulse)
            sleep(delay_period)

    stream = BytesIO()
    sleep(5)
```

The raspberry pi is

IV. Difficulties and Challenges

We faced many challenges during the completion of the project. We tried to do all the image processing on the raspberry pi however the pi, unfortunately, did not have either the necessary speed or space to handle the complicated image processing in OpenCV and Tensorflow. Therefore, we had to design an HTTP server using python to send the image to a Linux computer where the image is processed. Finding the right model for mask detection was also difficult. We tried several different frameworks before finding an efficient Tensorflow mask detection model from AIZOO. Remote work also proved difficult especially for a hardware project where only one of us could see the immediate results.

V. Market Evaluation

MaskIt has several market application. MaskIt could be used as an artificial bouncing mechanism to ensure that people use masks in public places. Large scale mask detection could also be useful in epidemiological and policy research.

VI. Suggested Improvements

MaskIt needs to be fully evaluated in larger public situations. If it is going to be useful in public places

VII. Conclusion

In conclusion, MaskIT has significant potential in research and commercial settings during the COVID pandemic. Given the lack of curret data on mask usage in practice and the importance of mask in the current public health it is important to create automated systems to tack account of mask usage. Some building may need automated gate systems to scan mask usage while universities will need to monitor the percentage of mask compliance on campuses. Systems like MaskIT will be of increasing public importance in the coming months.

References

[1] John Brandon. A doctor explains why 45protective mask, May 2020.
[2] Neil Vigdor. American airlines bans conservative activist who refused to wear a mask, Jun 2020.