West Visayas State University
College of Information and Communications Technology

FACE MASK AND PHYSICAL DISTANCING DETECTION SYSTEM USING COMPUTER VISION

THESIS SOFTWARE USER GUIDE

AUTHORS:
Christian T. Sarabia
Flora May D. Gicanal
Marianne Therese E. Tunggak
Hugo Leroy D. Chavez
Dr. Frank I. Elijorde

FACE MASK AND PHYSICAL DISTANCING DETECTION SYSTEM USING COMPUTER VISION

An Undergraduate Thesis

Presented to the Faculty of the

College of Information and Communications Technology

West Visayas State University

La Paz, Iloilo City

In Partial Fulfillment

of the Requirements for the Degree

Bachelor of Science in Computer Science

by

Hugo Leroy D. Chavez

Flora May D. Gicanal

Christian T. Sarabia

Marianne Therese E. Tunggak

January 2022

DISCLAIMER

This software project and its corresponding documentation entitled "*Face Mask and Physical Distancing Detection System using Computer Vision*" is submitted to the College of Information and Communications Technology, West Visayas State University, in partial fulfillment of the requirements for the degree, Bachelor of Science in Computer Science. It is the product of our own work, except where indicated text.

We hereby grant the College of Information and Communications Technology permission to freely use, publish in local or international journal/conferences, reproduce, or distribute publicly the paper and electronic copies of this software project and its corresponding documentation in whole or in part, provided that we are acknowledged.

Christian T. Sarabia

Flora May D. Gicanal

Marianne Therese E. Tunggak

Hugo Leroy D. Chavez

January 2022

Table of Contents

Getting Started

Face Mask and Physical Distancing Detection System using Computer Vision is a Research and Development Project which will be developed by the researchers to detect facemasks and physical distancing on individuals.

The system will utilize computer vision with a classification and prediction model to predict and analyze the data.

System Requirements

Software

- Anaconda for Spyder IDE
- Python 3.8
- TensorFlow 2.3.0
- GPU Drivers
    1. Nvidia CUDA 11.0
    2. cuDNN SDK 8.0.4

Hardware

- Operating System (Windows, Mac, Linux)
- Nvidia GPU with minimum 2GB VRAM
- Camera (Webcam, CCTV, Mobile Phone Client)

Installation

1. Download and install Anaconda Software which includes Spyder for the IDE and make sure add Anaconda to the PATH environment variable during the installation.

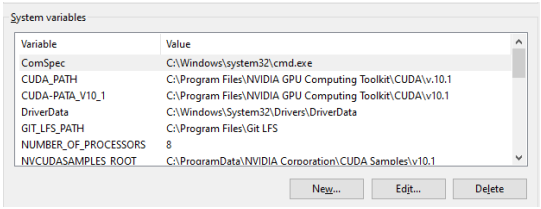
2. Once installed, create an environment to be used for this project along with Python 3.8 by opening Anaconda Prompt and type `conda create –name fmpdd python=3.8` where *FMPDD* stands for *Face Mask and Physical Distancing Detection*.

3. Then download the project software in this repository. This repository includes the necessary files for the project software to operate such as datasets for the training of the face mask detection model.

4. Once the project software is downloaded and extracted, set the current environment to *FMPDD* by using the command `conda activate fmpdd` and proceed to install all the required libraries in *requirement.txt* by using the command `pip install -r requirement.txt` inside the project's directory using Anaconda Prompt.

5. Download and install the specified GPU drivers;
    - Nvidia CUDA 10.1: (https://developer.nvidia.com/cuda-10.1-download-archive-base)
    - cuDNN SDK 8.0.4: (This requires to sign up an account to access the download link- https://developer.nvidia.com/cudnn-download-survey)

After downloading CUDA 10.1, add the folder `C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.1` as a system variable as shown in the picture below.

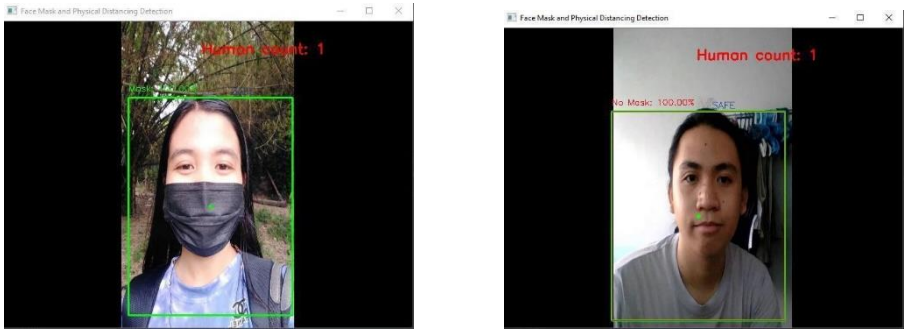


6. Download the Yolo Weights which is a pre-trained model to detect social distancing (https://pjreddie.com/media/files/yolov3.weights). From the "src "folder, create a folder named yolo-coco and paste the downloaded `yolov3.weights` file inside

7. Using Spyder IDE, to execute the main module, run *main.py*.

Usage

- Face Mask Detection- Once the system is running, a camera stream will be showing where it shows a box within a person's body and it will detect if the person if he/she is wearing a mask. If the person is wearing a mask, the box will become green with a label "Mask 100%", and if the person is not wearing a mask, the box will become red with a label "No Mask 100%".

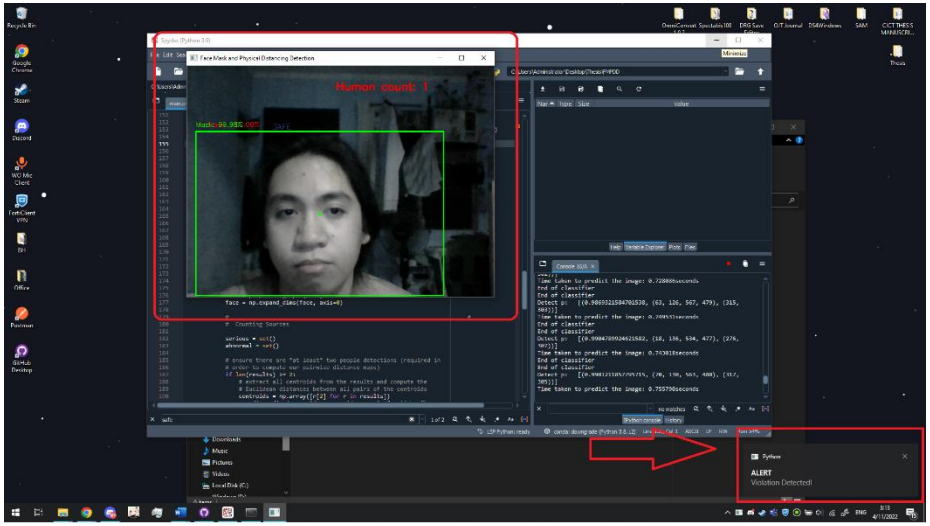


Person with a Mask vs Person without a Mask

- Physical Distancing Detection- Once the system is running, a camera stream will be showing where it shows a box within a person's body and it will detect if the person if he/she is following physical distancing. If the person is not with a nearby person, the box will become green with a label "SAFE" and if the person is with a nearby person, the box will become red with a label "NOT SAFE".

Note: Both of the features Face Mask Detection and Physical Distancing Detection has been merged so when the system detects an individual, it might show 2 boxes within the same range. Another scenario if the person is wearing a face mask but is nearby with a person, the face mask detection box will become green but the physical distancing detection box will become red.

- Alarm Notification System- If the system detects individuals who does not follow the health protocols, it will produce a beep then an automated audio where it says “Alert, Violation Detected, Please Follow Health Protocols”, and it will show a toast notification in the lower right screen of the user’s PC.



Troubleshooting for fixing possible bugs

- Slow/Stuttering camera stream- While running the software, you might encounter that the camera stream along with the detection is slow, this might be because of the absence of Nvidia GPU or the installation of OpenCV in your computer is not configured directly with CMake with CUDA backend so that will utilize your Nvidia GPU, otherwise it will use your CPU hence it is slow . But even so, the detection system works perfectly when regardless if the system is using the CPU or the GPU. Here is a link to a comprehensive tutorial if you want to configure OpenCV with CMAKE with CUDA backend. (https://medium.com/analytics-vidhya/build-opencv-from-source-with-cuda-for-gpu-access-on-windows-5cd0ce2b9b37). The figure below shows the log from the Spyder console in which the system used the CPU instead of GPU.

```
[ WARN:0] global C:\Users\appveyor\AppData\Local\Temp\1\pip-req-
build-9d_dfo3_\opencv\modules\dnn\src\dnn.cpp (1429)
cv::dnn::dnn4_v20200609::Net::Impl::setUpNet DNN module was not built with
CUDA backend; switching to CPU
```

- Random Hangs/Crashes while running- This error is encountered if the system was not able to detect any objects in the camera stream. This might due to several reasons such as external entities like the camera/webcam is being blocked by dust or any opaque material, camera/webcam is blurry, the area is dark or empty in which the system could not find any objects to detect. The solution is to make sure that the camera surface is clean and clear without obstructions, use a high-resolution/high-quality camera/webcam/CCTV, and confirm that the surveillance area is brightly lit. The figure below shows the error from the Spyder console where the system was not able to detect anything.

```
error: OpenCV(4.4.0) C:\Users\appveyor\AppData\Local\Temp\1\pip-req-
build-9d_dfo3_\opencv\modules\imgproc\src\color.cpp:182: error:
(-215:Assertion failed) !_src.empty() in function 'cv::cvtColor'
```

FAQs

(Frequently Asked Questions)

Q: Does the software works in other camera devices such as CCTVs to be used in establishments?
A: Yes. Any camera devices that can capture video data can be utilize by the software.

Q: May the face detection model be further trained to achieve higher accuracy?
A: Yes and No. You can train the model with datasets with both people wearing and not wearing face masks but the model has already achieved 99% accuracy so training the model is unnecessary.

Q: If I didn't configure my OpenCV with CMake, will I still able to use the software?
A: Yes. But you may able to experience a slow camera stream while the detection is running because the system will utilize your PC's CPU instead of its GPU.

Contact Details of the Development team

Hugo Leroy D. Chavez – hugoleroy.chavez@wvsu.edu.ph

Flora May D. Gicanal – floramay.gicanal@wvsu.edu.ph

Christian T. Sarabia – christian.sarabia@wvsu.edu.ph

Marianne Therese E. Tunggak – mariannetherese@wvsu.edu.ph