Enhancing Covid-19 Defense Matrix Parameters using AI Face Mask Detection Alarm System

Project in Introduction to Computer Vision

Jyresa Mae M. Amboang   
 *Bachelor of Science in Computer Application   
Mindanao State University Iligan Institute of Technology*   
Iligan City, Northern Mindanao, Philippines   
[jyresamae.amboang@g.msuiit.edu](mailto:jyresamae.amboang@g.msuiit.edu).ph

Crishelle A. Agopitac

*Bachelor of Science in Computer Applications*

*Mindanao State University Iligan Institute of Technology*

Iligan City, Northern Mindanao, Philippines

[crishelle.agopitac@g.msuiit.edu.ph](mailto:micamarie.moreno@g.msuiit.edu.ph)

Frances Nicole S. Gigataras  
 *Bachelor of Science in Computer Application   
Mindanao State University Iligan Institute of Technology*   
Iligan City, Northern Mindanao, Philippines   
[francesnicole.gigataras@g.msuiit.edu](mailto:jyresamae.amboang@g.msuiit.edu).ph

***Abstract***

**Covid-19 pandemic overwhelmed the global state with its alarming effects and mortality rate. As prevention, the World Health Organization (WHO) promulgated the compulsory use of face masks to mitigate its transmission. The researchers set out to create a detection model that assists the government in supervising the completion of the said health protocol. Programmed in Python, it determines individuals through the camera who are wearing masks and individuals without mask is detected, notifying the operator through the alarm system built in the program.**

**Keywords—Face Recognition, Face Detection, Face Mask, Algorithms**

1. INTRODUCTION

Face mask detection entails locating the face and assessing whether or not it is covered by a mask (Mohammed,2020). Face mask detection with an alarm system is one of the cutting-edge technologies that can assist in determining the number of persons wearing a face mask or without, and shielding a sizable population from the severity of the infection.

This paper provides a simple way to achieve this objective by utilizing some fundamental Machine Learning tools such as TensorFlow, Keras, OpenCV, Pandas, Scikit-Learn, etc. To protect people from the dangerous coronavirus infection within the vicinity, the researchers in this initiative offer the idea of face mask detection with an alarm system. Despite not being as advanced as you expected, this project represents a building block for future goals to establish face mask detection systems on campuses.

1. THEORY
2. *Some Fundamental Concepts*

The face mask detection system indicated in this research can monitor and detect mask use from camera feeds and alert when a violation occurs. This initiative is beneficial, especially in congested locations.

Face mask detection systems have expanded their usefulness beyond their original scope as computer applications to any technology that has a camera. Using algorithms and deep learning technology, face mask detection is an Artificial Intelligence (AI) analytical tool that can discern the difference between a person wearing a face mask and one who isn't (Ouellette,2023). By using an existing established camera network to recognize face masks in real-time, AI as a tool can assure the wearing of face masks in public spaces.

1. *Brief History*

The COVID-19 pandemic was first brought to light in China in December 2019. On March 11, 2020, the World Health Organization (WHO) designated the illness that's brought on by the SARS-CoV-2 virus becoming a global pandemic. To enforce the usage of face masks in public spaces, scientists have tried to construct automatic facial mask recognition systems. Other researchers created their own methods for observing face masks in public places after the COVID-19 epidemic. Using image processing algorithms and surveillance systems to monitor public spaces and guarantee that no one's face is visible in crowded locations. Researchers and scientists have found that face masks can limit the rise of COVID-19 in public areas. They proposed a new method to determine the level of face mask usage. Two sorts of face mask conditions exist: mask detected and no mask.

1. *Applications to Society*

The World Health Organization (WHO) suggested that people in public places wear face masks at all times. The issue with enforcing these and other pertinent safety regulations is that citizens are unwilling to do so, and relevant organizations are unable to monitor and enforce compliance. This research investigates the creation of camera-enabled face mask recognition software that will simplify the oversight and application of this protocol.

The main objective of this research is to develop some powerful technology to stop the Coronavirus from spreading.The researchers aim to enhance the monitoring system of wearing face masks through a face mask detection program. It specifically aims to:

* to design a Python-program monitoring system
* to code a no-face mask recognition program
* to embed no-face mask recognition with an alarm code

1. *SYSTEM ANALYSIS*

In their study on face mask detection systems, Dr. Ganesh and Balapriya emphasized that while many detector systems are made globally, science still needs optimization; a higher, more accurate detector is required. If we estimate the cost of the system to implement it, there won't be one because all that is needed is a camera. Based on neural networks, this system. Several methods or algorithms are listed below:

* *EXISTING SYSTEM*

1. Deep Convolutional Neural Network (DCNN/CNN)

CNN, which draws its inspiration from biological visual systems, has lately been utilized to replicate human vision and to recognize and classify patterns in pictures and videos of people's faces.

1. Artificial Neural Networks(ANN/ABANN)

This technique, which is a subset of machine learning and is at the core of deep learning algorithms, processes labeled faces and alignment information that has been identified in the system. Performance may be slow but detection rates are still high.

1. Recurrent Neural Networks(RNN)

RNN is a technique for face verification in computer vision applications that predicts the results of a given sequence in the future. This method utilizes geometric face verification features such as visual modalities, including intensity gradient, skin color models, and geometric face construction.

* *PROPOSED SYSTEM*

The other well-known algorithms listed below are primarily used in face detection but are also widely used in face mask recognition systems. This proposed system focuses on how to identify the person on an image or video stream wearing a face mask or not with the aid of neural networks and packages, and computer vision by using:

1. Open Source Computer Vision Library (Open CV)

OpenCV is an open-source library used in computer vision and machine learning software libraries mainly used in the system to process images and videos in identifying the faces.

1. Tensorflow

A general-purpose machine learning library that is mostly used in Deep Learning Framework, an open-source artificial intelligence library for making machine learning and creating neural networks faster and easier than readily aided in picture categorization.

1. Keras

It provides clear and actionable feedback upon user error for aligning faces on input images that evaluates the small subset of the LFW (Labeled Faces in the Wild) dataset, which you can replace with your custom data with images like with mask or no mask dataset. Open-source python library made for developing and evaluating CNN and OpenCV on input images.

1. Numpy

Mathematical operations on an array are utilized by MobileNetV2 to quickly and efficiently extract data when converting photos into an array format so that we can store the trained model.

1. Imutils

Alternative in using OpenCV library, this is helpful when we are shifting multiple images and in multiple directions like image translation, rotation, resizing, skeletonization, or blur amount detection.

1. Scikit-learn (Sklearn)

Users can use an open-source machine learning library to implement machine learning in Python, which is well-known for its ability to support face recognition and motion detection. The library provides several tools for model fitting, data preparation, model selection, model assessment, and other functions.

1. MatPlotlib

This is essential for building a visual data plot in the cross-platform, data visualization, and graphical charting library, as well as its numerical extensions NumPy, and Pandas, which alter elements of a figure or picture.

1. MobileNetV2

The core of this architecture, which was made by Google to do real-time mask recognition and was trained and tested using two separate datasets of masks, can manage light and low processing workloads. It is similar to CNN in that it processes pictures through many layers.

1. Pandas

One of the tools for machine learning is used to clean, analyze, explore, manipulate, and visualize data. This software package, one of the predominant libraries for Python programming, includes data structures and is used for data manipulation and analysis.

1. Software Requirements

The researchers functional and non-functional requirements are laid out here for the software system. They outline our operating system and the system's tools, such as:

Operating System: Any Operating System with Internet

Connectivity

Language: Python 3.10.0

Packages Required:

* Tensorflow>= 3.10
* numpy
* imutils
* keras
* opencv-python
* sklearn
* pandas
* matplotlib

1. Hardware Requirements

This is technical description of the computer’s components and capabilities such as processor speed, model, and manufacturer, etc. these are:

Processor: Intel i3

Hard Disk: 1TB

RAM: 4-8GB

1. Python

Face recognition consists of thousands of lines and characteristics that must match. The tasks of face recognition, motion detection, and image processing may be separated using this programming language and tools. Additionally, this enables language flexibility, simplicity, and ease of recognition.

1. METHODS

In this project, face mask identification is achieved using an Optimistic Convolution Network that can detect whether or not people are wearing masks in public by observing them automatically. In order to determine whether someone is wearing a face mask, our system combines the Convolutional Neural Network model with the TensorFlow and Keras algorithms.

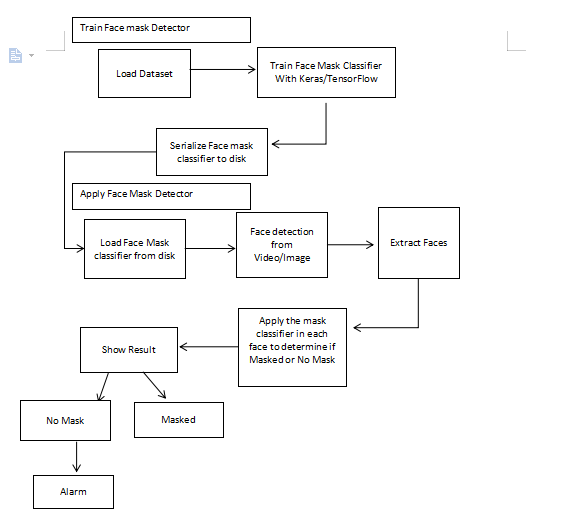
The figure below represents the flow of our project.

Fig 3.1 Face Mask Detection Alarm System Diagram

PHASE 1: TRAINING

First, the researchers obtained a data set from Kaggle and used it to train the system using Keras and TensorFlow; once that is complete, we will load the face mask classifier from disk.In order to train large amount of images and classify images with higher quality we additionally use MobileNet.

PHASE 2 : DEPLOYMENT

After loading the picture dataset from Keras, it will

turned into an array, after which MobileNet will process the input image and append it to the data list. With the help of MobileNet and OpenCV the proposed system face mask detection that can be done in Real Time .As a result the video will display a person with a square bound box either green for masked and red for no mask. This system will monitor continuously, and whenever a person is identified without a mask then the alarm will sound.

1. CONCEPTUAL FRAMEWORK

The researchers used Python IDLE and Visual Studio Code to create the program for the face mask detection alarm system. Installing programming-related modules, packages, and libraries like OpenCV-python, NumPy, keras, sklearn, etc., involves using the Command Prompt. The Python file was changed into an executable (EXE) file using Command Prompt so it could run from an application form. The image below serves as a representation of the conceptual framework for this project.



Fig. 4.1 Face Mask Detection Framework

It is proposed to design a system that is capable of identifying a person’s face, even if it is with or without a mask. Two databases must be used for the system to operate effectively: the first database is used for classifier training and contains a sizable number of photographs of persons not wearing face masks. The second photographs of persons with biosafety items are present (face masks) in the detection system. To have higher precision and resilience, MobileNetV2 is the architecture employed, and the input data are received either from an image or a video.

1. System Implementation
2. *Assembling the image dataset*

The training and the model-building process uses a dataset of almost 2000 photos. As our training dataset contains many images, we would begin by plotting that fell into the most categories that we could find. The database contains around 600 pictures of people wearing face masks and 600 pictures of people without face masks, both of which have the designation "with\_mask" or "without\_mask" respectively on the dataset folder.

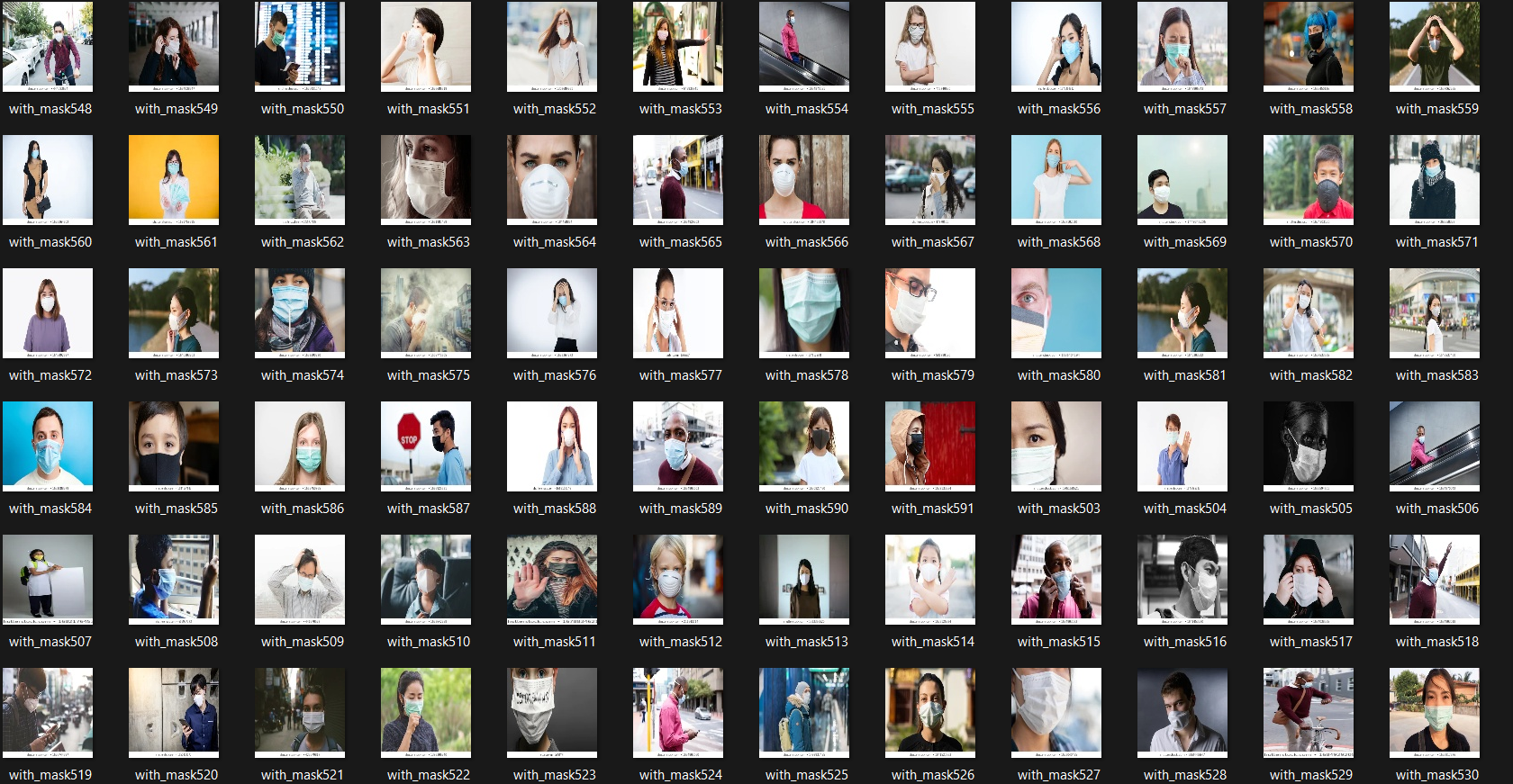


Fig.4.2 with\_mask

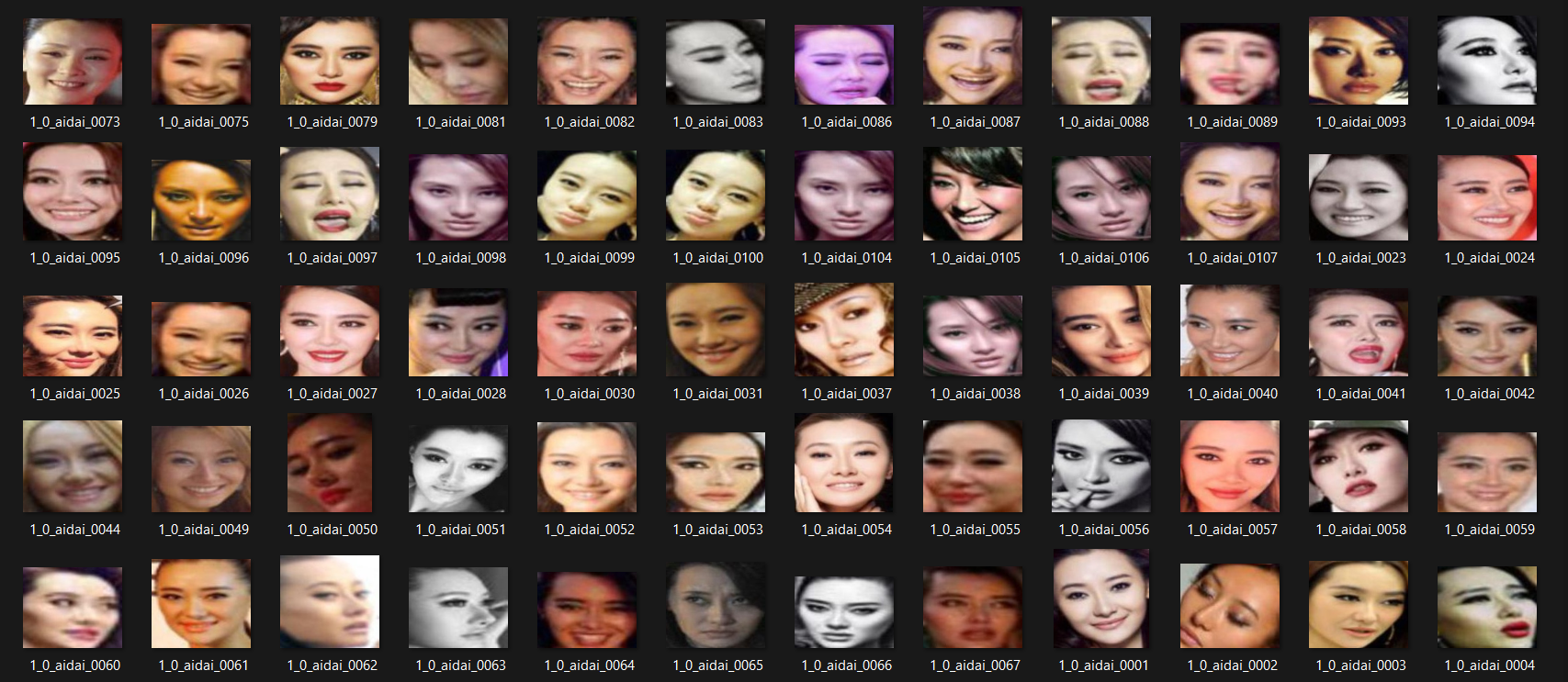


Fig.4.3 without\_mask

1. *Image Processing*

Technical analysis of a picture utilizing sophisticated algorithms is referred to as image processing. In this case, the image serves as the output of which is useful information. To manipulate and process images, use ***Numpy***. For image processing, ***Scikit-learn*** offers a wide range of algorithms. The real-time computer vision-focused image processing toolkit ***OpenCV*** is utilized in a variety of fields, including 2D and 3D feature toolkits, facial & gesture recognition, Human-computer interaction, mobile robots, object identification, and others.

1. *Producing the MobileNetV2 Classifier*

* MobileNets are a type of Convolutional neural network where images are processed through different layers and for the base model, we use imagenet and give input shape as 224x224 with 3 channels for colored images(RGB)
* The pretrained network can classify images around 2000 with persons that have face masks and those who do not. As a result, the network has learned rich feature representations for a wide range of images.

1. *Training the dataset using Tensorflow and saving it in H5 Format*

The hierarchical data format, or H5, is a multidimensional array that may carry the data of the detection system, and it is used to train the system using the command prompt by specifying the batch size.

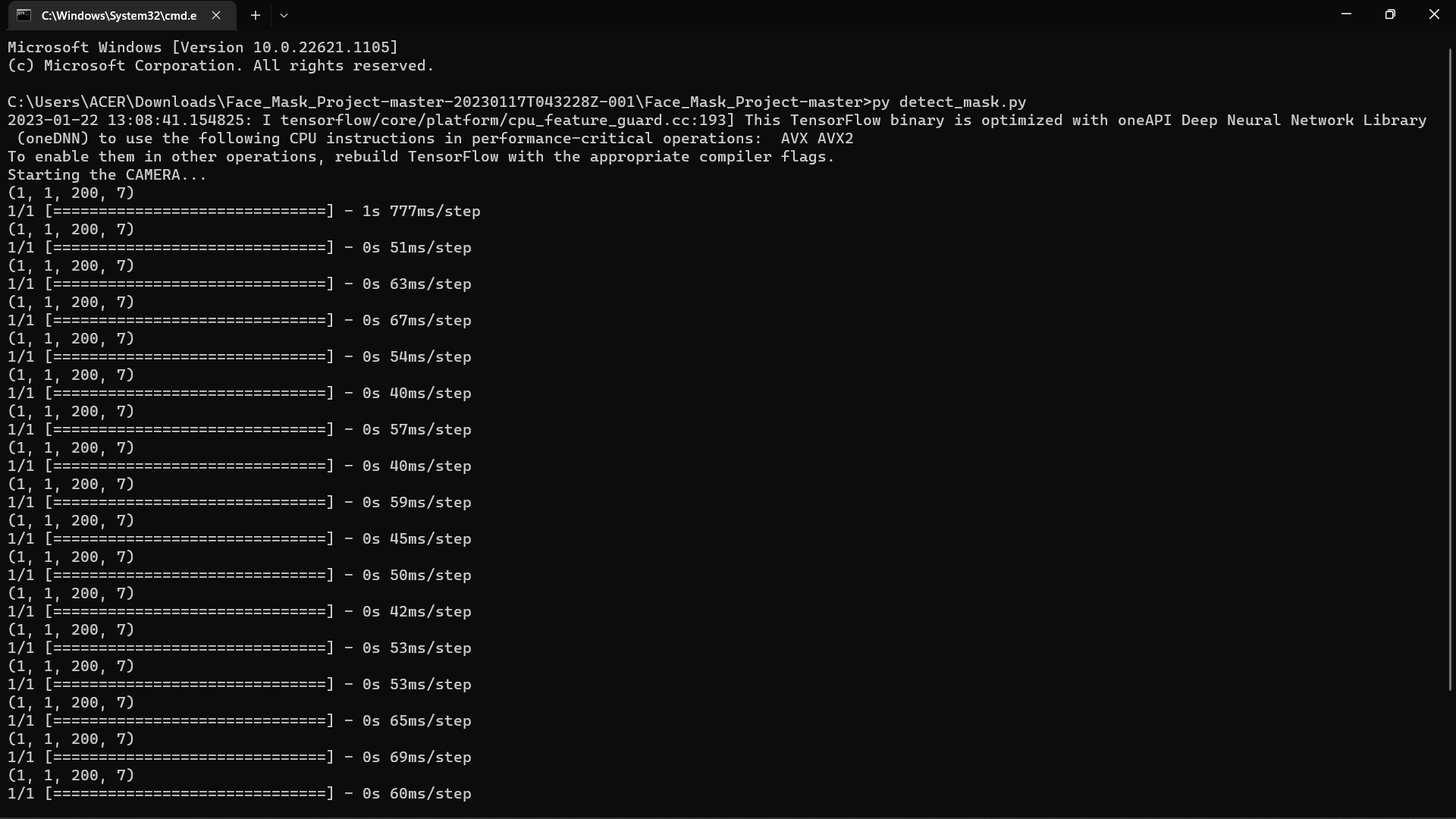


Fig.4.4 Command Prompt Terminal

1. RESULTS AND DISCUSSION
2. *TESTING*

| **ACCURACY LEVEL** | |
| --- | --- |
| No. of Tries | Percentage Rate of Accuracy Level |
| 1 | 67 |
| 2 | 77 |
| 3 | 45 |
| 4 | 50 |
| 5 | 71 |
| 6 | 80 |
| 7 | 75 |
| 8 | 82 |
| 9 | 75 |
| 10 | 89 |
| **TOTAL** | **71%** |

The researchers attempted to test the system more than ten times, notably in terms of accuracy, as part of the requirements for this particular project. After a considerable amount of trials, we found that the level of accuracy is 71%. Although the program automatically detects the scanned faces, there is a flaw in the system that makes it difficult for the system to determine whether a certain individual is wearing a mask or not, which leads to erroneous alarm behavior. For the program to quickly and precisely detect and recognize the face scanned on the camera, each individual to be scanned must remain stationary for a few seconds.

1. *CODES*

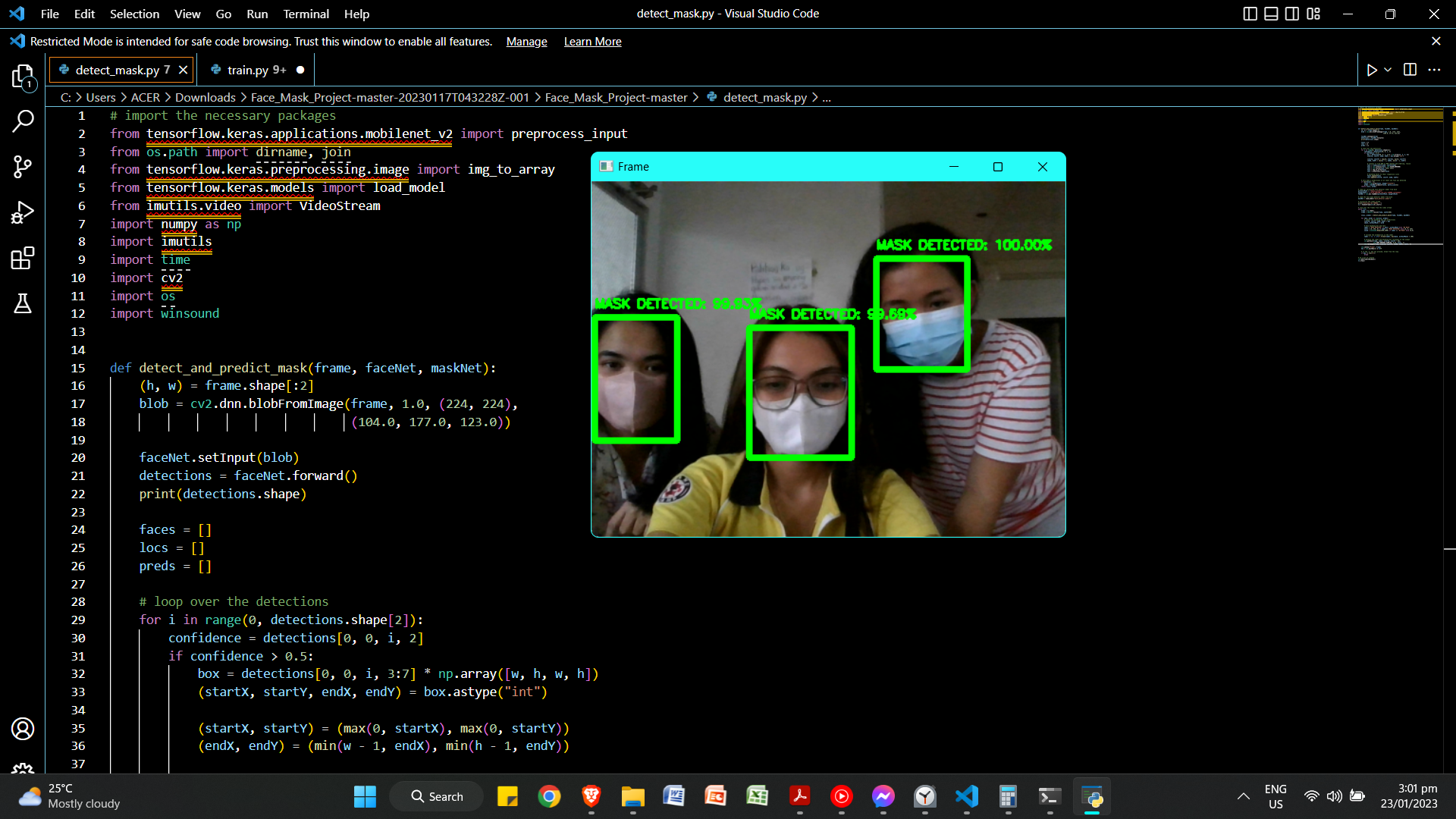
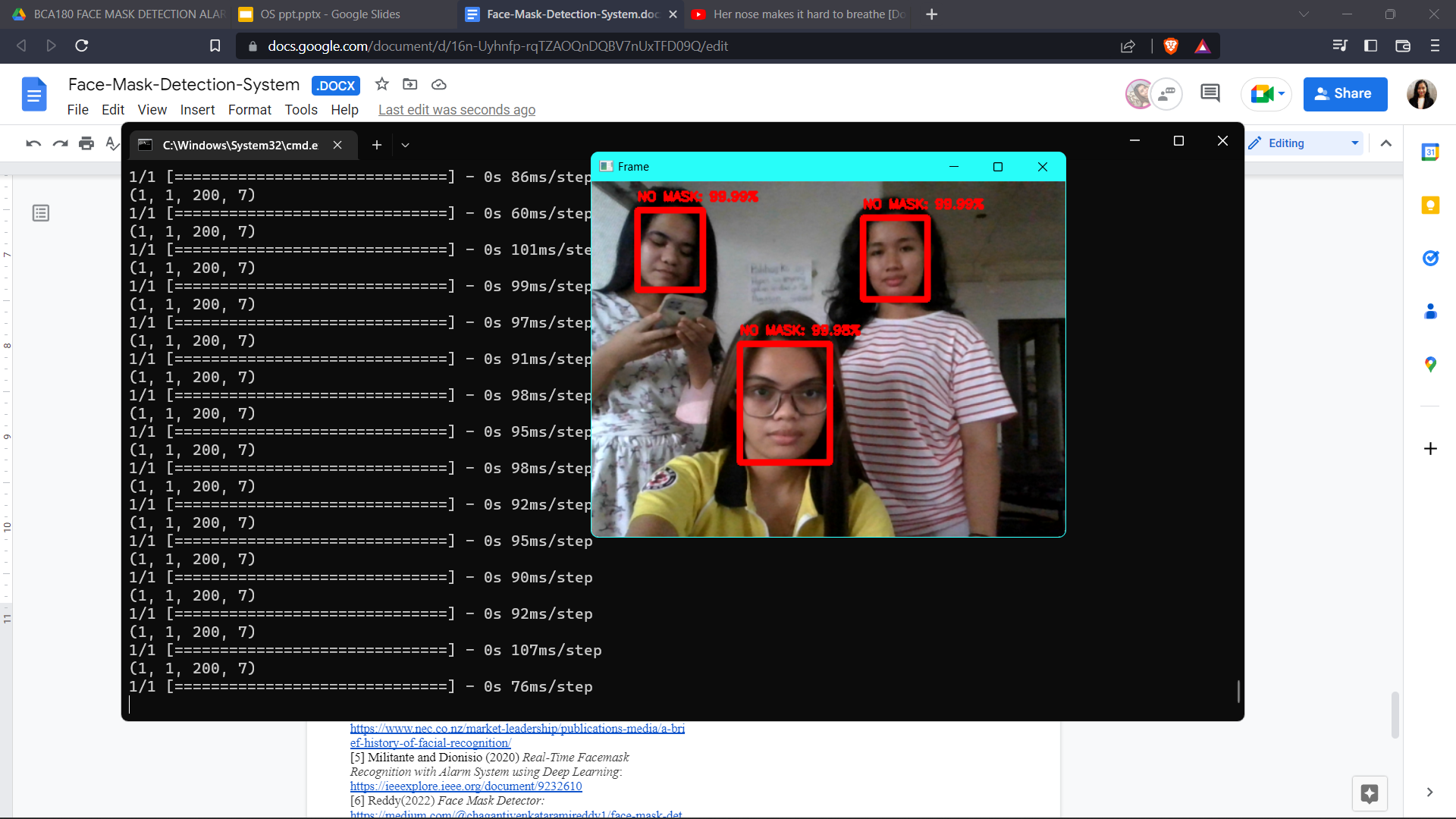
| from tensorflow.keras.applications.mobilenet\_v2 import preprocess\_input  from os.path import dirname, join  from tensorflow.keras.preprocessing.image import img\_to\_array  from tensorflow.keras.models import load\_model  from imutils.video import VideoStream  import numpy as np  import imutils  import time  import cv2  import os  import winsound  def detect\_and\_predict\_mask(frame, faceNet, maskNet):  (h, w) = frame.shape[:2]  blob = cv2.dnn.blobFromImage(frame, 1.0, (224, 224),  (104.0, 177.0, 123.0))  faceNet.setInput(blob)  detections = faceNet.forward()  print(detections.shape)  faces = []  locs = []  preds = []  # loop over the detections  for i in range(0, detections.shape[2]):  confidence = detections[0, 0, i, 2]  if confidence > 0.5:  box = detections[0, 0, i, 3:7] \* np.array([w, h, w, h])  (startX, startY, endX, endY) = box.astype("int")  (startX, startY) = (max(0, startX), max(0, startY))  (endX, endY) = (min(w - 1, endX), min(h - 1, endY))  # convert it from BGR to RGB channel and ordering, resize  face = frame[startY:endY, startX:endX]  face = cv2.cvtColor(face, cv2.COLOR\_BGR2RGB)  face = cv2.resize(face, (224, 224))  face = img\_to\_array(face)  face = preprocess\_input(face)  # bounding boxes to their respective lists  faces.append(face)  locs.append((startX, startY, endX, endY))  # only make a predictions if at least one face was detected  if len(faces) > 0:  faces = np.array(faces, dtype="float32")  preds = maskNet.predict(faces, batch\_size=32)  return (locs, preds)  # load serialized face detector model from disk  prototxtPath = r"deploy.protect"  weightsPath = r"res10\_300x300\_ssd\_iter\_140000.caffemodel"  faceNet = cv2.dnn.readNet(prototxtPath, weightsPath)  # load the face mask detector model from disk  maskNet = load\_model("mask\_detector.model")  # initialize the video stream  print("Starting the CAM...")  vs = VideoStream(src=0).start()  # loop over the frames from the video stream  while True:  frame = vs.read()  frame = imutils.resize(frame, width=400)  (locs, preds) = detect\_and\_predict\_mask(frame, faceNet, maskNet)  for (box, pred) in zip(locs, preds):  # unpack the bounding box and predictions  (startX, startY, endX, endY) = box  (mask, withoutMask) = pred  # draw bounding box text and and sound  label = "MASK DETECTED" if mask > withoutMask else "NO MASK"  color = (0, 255, 0) if label == "MASK DETECTED" else (0, 0, 255)  sound = winsound.Beep(2500,1000) if label == "NO MASK" else (0,0)  # include the probability in the label  label = "{}: {:.2f}%".format(label, max(mask, withoutMask) \* 100)  # display the label and bounding box rectangle on the output  cv2.putText(frame, label, (startX, startY - 10),  cv2.FONT\_HERSHEY\_SIMPLEX, 0.45, color, 2)  cv2.rectangle(frame, (startX, startY), (endX, endY), color, 5)  cv2.imshow("Frame", frame)  key = cv2.waitKey(1) & 0xFF  # if the `q` key was pressed, break from the loop  if key == ord("q"):  break  # do a bit of cleanup  cv2.destroyAllWindows()  vs.stop() |
| --- |

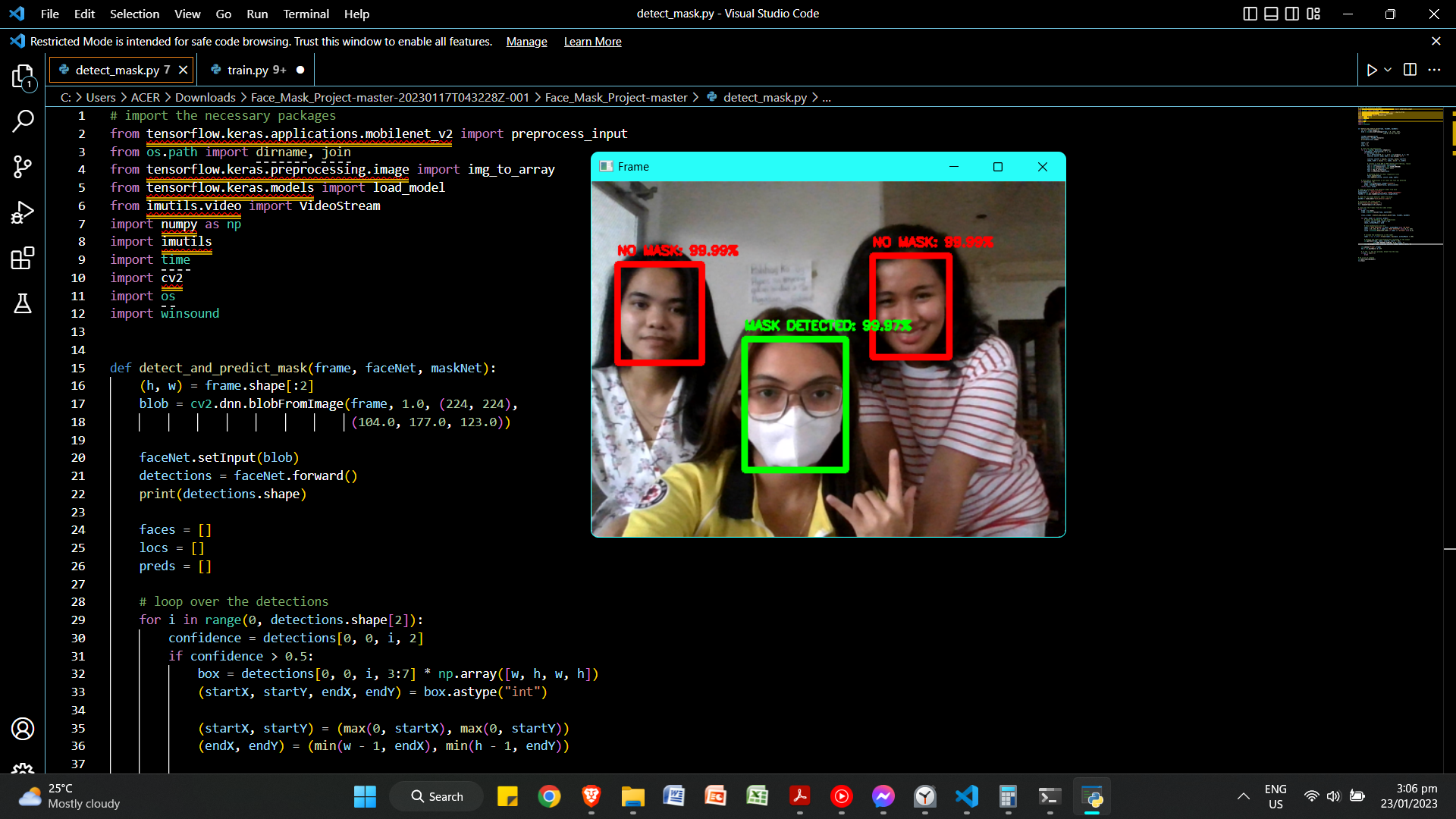
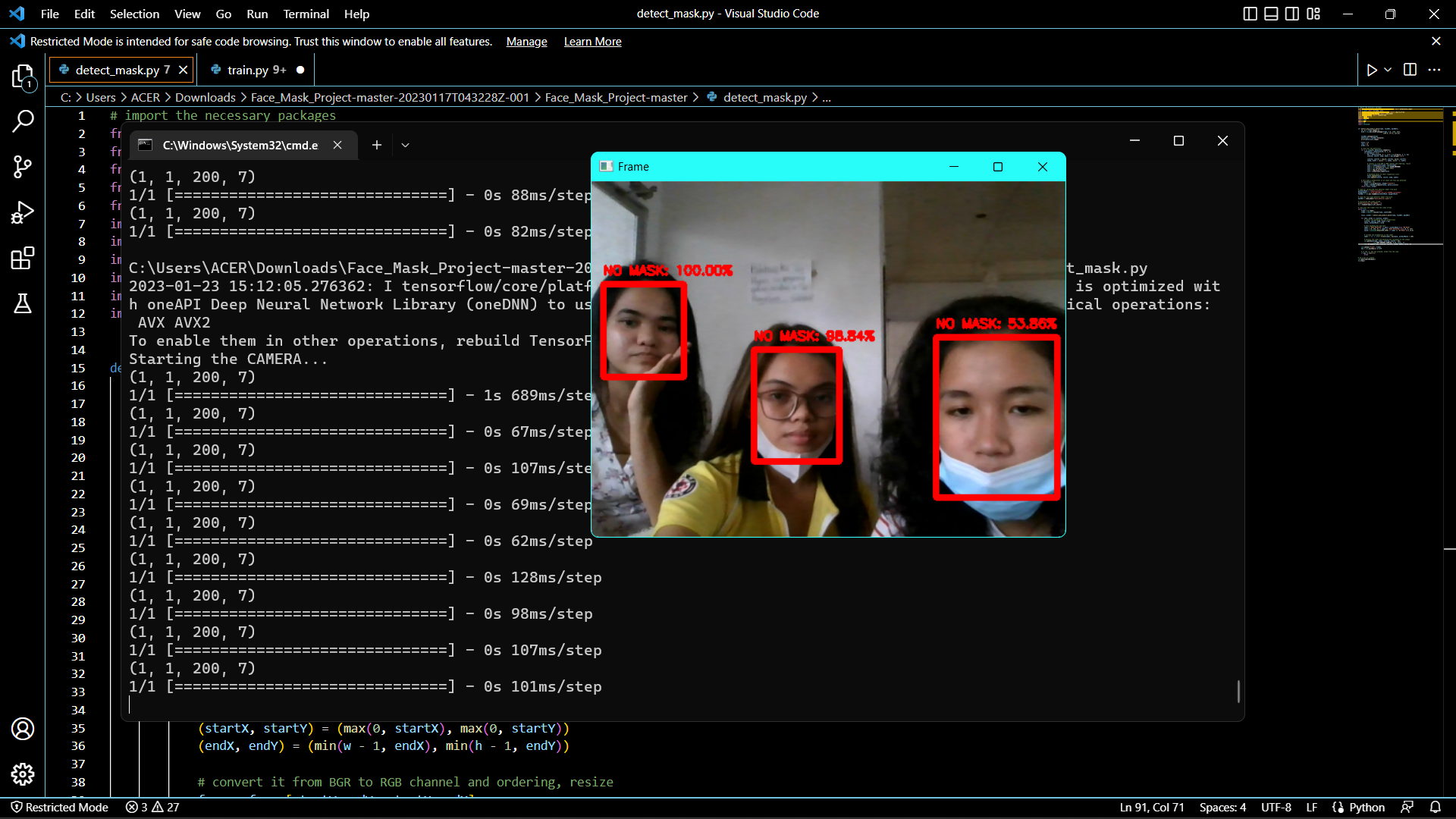
If you want to access the whole code, you can click [here](https://drive.google.com/drive/folders/1nWsqMNOiue1bYed_SV1x3bV41XnUbf_t?usp=share_link) or the link below.

<https://drive.google.com/drive/folders/1nWsqMNOiue1bYed_SV1x3bV41XnUbf_t?usp=share_link>

1. CONCLUSIONS

The images below serves as the output of our project:



The examples of this project's output are shown in the pictures above. You may access further output through the video that is included with the project's files.

The researchers' results revealed that the no face mask recognition system encoded has an accuracy rate of 71%. According to studies, an accuracy of above 50% indicates that the device/program is effective for use. Since it is 71%>50%, the program then satisfies this standard. Since the system is so sensitive to even little movements of the subject of the camera, there are times the alarm will function abnormally. The system also cannot detect if the person is wearing a mask if he/she is far from the camera. So the researchers concluded that the system is not 100% accurate and this project needs more improvements

Once again, the researchers will take consideration in developing this initiative for future beneficial usage on campus and even for the entire society. For more information regarding the project’s development, please proceed to the RECOMMENDATION section in this paper.

VI. RECOMMENDATION

The researchers on this specific study are thinking about upgrading and using their work on a larger scale, as well as recommending it to future researchers.

This idea can be applied in face mask detection systems on campuses and in other places where it can be very helpful in determining whether or not the individual is wearing a face mask. The alarm will blare and the operator will be able to determine who it was once there is no face mask detected.

The better would be to even attempt to use it on a more complex level. As it can recognize and identify every resident, including those who wear face masks or not, it may be employed for technical advancement, particularly in the city's security against the spread of the virus

For further studies, the researchers would like to recommend the following: applied in terms of monitoring the security of an individual against the spread of Covid19 and other viruses in the near future, enhancing motion sensor cameras, and identifying what type of mask the person is wearing. This can be used in the entrance gate of a particular place such as campuses, malls and other establishments and checks every individual passing by the camera and detects whether they are wearing a mask or not. But for now it will only be applicable in indoor places where a person should face the camera closely.

REFERENCES

[1] Abhishree, Latha, Manikantan, and Ramachandran (2015) *Face Recognition Using Gabor Filter Based Feature Extraction with Anisotropic Diffusion as a Pre-processing Technique*: <https://reader.elsevier.com/reader/sd/pii/S1877050915003853?token=7544DF0837C84D44F524C4871FDEB8816441BFF6EFC4011D6D6C95C54689DEE361017B81E315A6DE6EB467F62248CA28&originRegion=us-east-1&originCreation=20230122122100>

[2] Triantafyllidou and Tefas (2016) *Face detection based on deep convolutional neural networks exploiting incremental facial part learning*: <https://ieeexplore.ieee.org/document/7900186>

[3]Shang, D., & Turk, D. (2008). Eigenfaces. Retrieved from Scholarpedia: <http://www.scholarpedia.org/article/Eigenfaces#Face_recognition>

[4] *A brief history of Facial Recognition*. (2020, May 26). Retrieved from NEC: <https://www.nec.co.nz/market-leadership/publications-media/a-brief-history-of-facial-recognition/>

[5] Militante and Dionisio (2020) *Real-Time Facemask Recognition with Alarm System using Deep Learning*: <https://ieeexplore.ieee.org/document/9232610>

[6] Reddy(2022) *Face Mask Detector:* <https://medium.com/@chagantivenkataramireddy1/face-mask-detector-c75aee8cd932>

[7] Hussain, Ismail, Alroobaea, & Ullah(04 May 2022) *Face Mask Detection Using Deep Convolutional Neural Network and MobileNetV2-Based Transfer Learning,* Nawab Muhammad Faseeh Qureshi: <https://www.hindawi.com/journals/wcmc/2022/1536318/>

[8] Tam(30 October 2021) *Face Recognition using Principal Component Analysis*, Machine Learning Mastery:<https://machinelearningmastery.com/face-recognition-using-principal-component-analysis/>

[9]Asghar, Muhammad Zubair, et al. “*Facial Mask Detection Using Depthwise Separable Convolutional Neural Network Model During COVID-19 Pandemic.”* Frontiers, 31 Jan. 2022, www.frontiersin.org/articles/10.3389/fpubh.2022.855254/ful

[10]“Kaggle: Your Machine Learning and Data Science Community.” *Kaggle: Your Machine Learning and Data Science Community*, [www.kaggle.com](http://www.kaggle.com)

Accessed 23 Jan. 2023.

[11]Mitrefinch, ”*How to ensure accuracy in data collection processes”*[https://www.mitrefinch.ca/blog/techology/how-to-ensure-accuracy-in-data-collection-processes](https://www.mitrefinch.ca/blog/technology/how-to-ensure-accuracy-in-data-collection-processes), Accessed 2020.