**CLASS ATTENDANCE & FACE MASK COMPLIANCE DETECTION SOFTWARE**

**User Manual**

Version 1.0

**Authors**

David Weir (19433086)

Cian Mullarkey (19763555)

**Supervisor**

Hossein Javidnia

**Date Finished**

03/03/2022

**Abstract**

The general area covered by this project is that of machine vision, in particular facial recognition. The aim of the project is to achieve a functioning classroom attendance log using a camera and a facial recognition algorithm to identify and then log and record students who have attended the particular class. The project is also capable of noting whether a student is wearing a face mask and making note of this along with their attendance.

**Table of Contents**

[**Table of Contents** 2](#_Toc97135409)

[**USER MANUAL** 3](#_Toc97135410)

[**1.** **Introduction** 3](#_Toc97135411)

[**1.1 Purpose & Scope** 3](#_Toc97135412)

[**1.2** **Primary Business Functions** 3](#_Toc97135413)

[**1.3 Glossary** 3](#_Toc97135414)

[**2.** **System Capabilities** 4](#_Toc97135415)

[**2.1** **Purpose** 4](#_Toc97135416)

[**2.2** **General Description** 4](#_Toc97135417)

[**3.** **System Functions** 5](#_Toc97135418)

[**3.1 Generate New Dataset** 5](#_Toc97135419)

[**3.2 Feature Extraction** 6](#_Toc97135420)

[**3.3 Attendance** 7](#_Toc97135421)

[**4.** **Operating Instructions** 7](#_Toc97135422)

[**3.2** **Initiate Operation** 7](#_Toc97135423)

[**3.3** **Terminate & Restart Operations** 7](#_Toc97135424)

[**5.** **Error Handling** 8](#_Toc97135425)

# **USER MANUAL**

## **Introduction**

### **1.1 Purpose & Scope**

This User Manual contains all essential information required by the user to operate and maintain our designed system. This manual includes a description of all system functions and capabilities, modes of operation, error handling, step-by-step procedures for system access and use.

### **Primary Business Functions**

The Class Attendance & Mask Compliance system offers 3 primary functions to the user. These are:

* **Dataset Generation:** Allows the user to capture face images of a student from the webcam and store them for use in face recognition during attendance taking. Allows the users to add new students to the dataset of students for recognition.
* **Feature Extraction:** Creates a model for use during attendance taking. It only needs to be run once after installation and anytime a student has been added or removed from the database.
* **Attendance Taking:** Opens the user’s camera and marks the attendance, initial time of attendance and mask compliance of all students that can be recognized. This data is stored in a CSV file called “Rollbook \_\_/\_\_/\_\_ (today’s date)”.

### **1.3 Glossary**

**Dataset:** A folder containing a number of subfolders storing the images of students and named after the relevant student.

**Feature Extraction:** The process of extracting face component features (e.g. eyes, nose, mouth) from the image of a human face.

**CSV:** A Comma-Separated Values file is a comma delimited text file where each line is a data record.

**GUI:** A graphical user interface, a menu that allows the user to control the system and its functions.

**CMD:** Command-line interpreter.

## **System Capabilities**

### **Purpose**

This system is designed to quickly and easily take the attendance of a class/lecture by using face recognition to recognize students and mark their attendance as present. The system also tracks the student’s mask compliance (i.e. are they wearing a face mask).

### **General Description**

The system is controlled by using our GUI to select the different functions available to the user. Seen here:

Diagram

Description automatically generated

To begin the system operation, we simply run the command “python gui.py” from a CMD terminal.



We can also see above our 3 main functions: Generate New Dataset, Extract Features and Attendance. We can use these buttons to begin running one of these functions.

## **System Functions**

### **3.1 Generate New Dataset**

#### **3.1.1 Function Description**

**Purpose and Scope:** This function focuses in on a user’s face and captures a set of images. These images are stored in a subfolder of the “dataset” folder and should be named after the person whose images are stored within it.

This function is used when adding a new student to the class. The system requires these images to recognize the student at a later date when it is asked to take attendance.

**Initialization & Operation:** To initialize the function, the user must input both a first and last name, used in the naming of the subfolder that will be created. As seen in the example below:

Diagram

Description automatically generated

Once these are inputted, simply click the Generate New Dataset button and a new window will open confirming that you wish to generate this dataset.

Graphical user interface

Description automatically generated

Following this, a window will appear in order to capture the images.

**Expected Results:** We expect our results to be a new subfolder within the “dataset” parent folder with the same naming as the first and last names that were inputted. Inside the new folder we expect to see at least 30 images of the student.

#### **Results**

After running, we should see a new folder inside the parent “dataset” folder with the name we inputted. We see an example of this below:

Before running Generate New Dataset:

Graphical user interface, text, application

Description automatically generated

After running Generate New Dataset:

Graphical user interface, text, application

Description automatically generated

Inside each of these folders is a set of images of the student for which the folder is named.

### **3.2 Feature Extraction**

#### **3.2.1 Function Description**

**Purpose and Scope:** The Feature Extraction function only ever needs to be run after first installation and whenever a student’s subfolder is added or deleted from the datasets folder.

This function is run on the images we have created using the Generate New Dataset function. It is used to create a model of face embeddings for each person that we can use for the facial recognition during attendance taking.

It extracts the face component features (e.g. eyes, nose, mouth) from the images of the students’ faces creating a set of embeddings / dimensional vectors for each face.

**Initialization & Operation:** To begin running the feature extraction program, click the “Extract Features” button on the menu and a confirmation box will appear.

Graphical user interface, text, application

Description automatically generated

This function does not require any user input before or during it’s operation. Once the confirmation button is clicked it takes the dataset as input automatically.

**Expected Results:** After running the extract feature function, we expect to see a new file created called “face\_enc”, which we can see below:

This completes the functions operation and the user does not have to interact with the new file.

#### **3.2.2 Results**

Once the Feature Extraction function has completed it’s operations, we see a new file called “face\_enc” is created in the same location as the “gui.py” file. We can see an example of this file below:



The result of this function (face\_enc file) is used during the course of running our Attendance function.

### **3.3 Attendance**

#### **Function Description**

**Purpose and Scope:** The attendance function is the main purpose function of the application and works based on the running of the two previous functions.

**Initialization & Operation:**

**Expected Results:**

#### **Results**

## **Operating Instructions**

### **Initiate Operation**

To begin the system operation, execute the gui.py Python file from your command prompt. This will begin running our user interface menu, from which you can access all of our system functions.

### **Terminate & Restart Operations**

## **Error Handling**