

**PROJECT DOCUMENTATION**

REG NUMBER**: R206658B**

NAME**: CHARLES PANASHE MUPEMHI**

FACULTY: **COMPUTER ENGINEERING INFORMATICS AND COMMUNICATION**

DEPARTMENT**: COMPUTER SCIENCE**

PROJECT TITLE**: FACIAL RECOGNITION BASED ATTENDANCE SYSTEM**

**INTRODUCTION**

Automated attendance systems using facial detection and recognition technology have been developed and implemented in various environments such as educational institutions, workplaces and event venues. These systems utilize biometric data, specifically facial features to identify individuals and monitor their attendance, offering higher accuracy and efficiency compared to traditional approaches like sign-in sheets or roll calls.

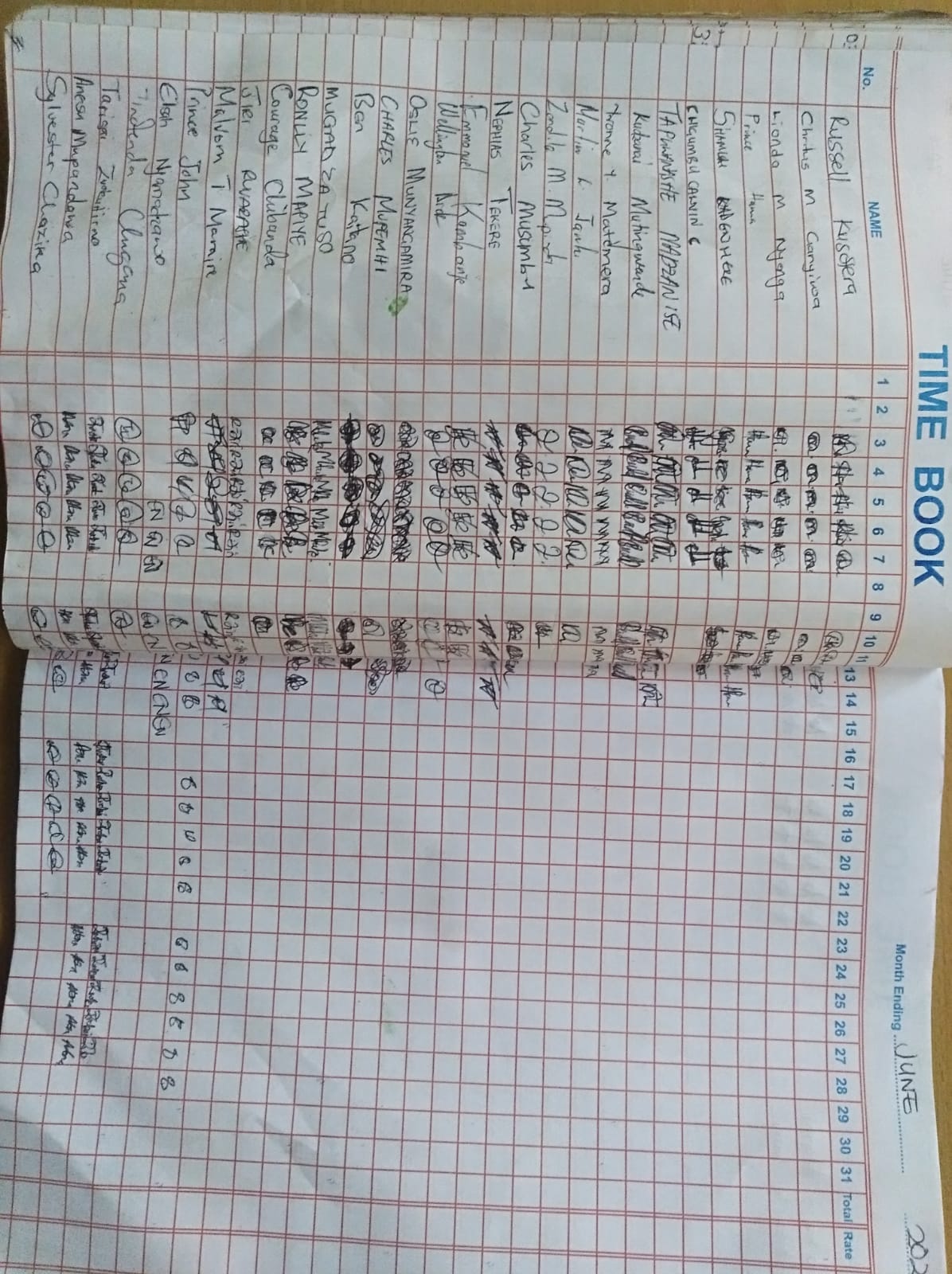
One proposed system for automated attendance tracking, for example, at a work place, captures members’ images upon arrival and employs a facial recognition algorithm to identify and record their attendance. This system is designed to be user-friendly and efficient, utilizing commonly available devices.

**BACKGROUND**

Facial recognition attendance systems are a recent innovation in attendance management that leverage the power of artificial intelligence (AI) and facial recognition technology. These systems streamline the process of recording employee or student attendance by automatically identifying individuals through facial features. This system gains popularity in various workplaces and educational institutions due to their efficiency, accuracy, and hygiene benefits. However, it's important to consider potential privacy concerns and ensure proper data security measures are in place.

**AIM**

The primary aim of facial recognition attendance systems is to revolutionize attendance management by offering a more efficient and automated process as they automate the task of marking attendance, eliminating the need for manual processes like paper sign-in sheets. This saves time for administrators and allows them to focus on other tasks. The system is also highly accurate as it minimizes errors associated with traditional methods, such as buddy punching or attendance sheets being tampered with.



There is also the issue of better hygiene, especially in environments where hygiene is a priority, such as healthcare facilities and when there is an outbreak like covid or typhoid, there will be no need to touch the pen and sign in your attendance but the smart system will automate everything.

**MOTIVATION**

Ever since I started working in the Harare City council, I felt the pain of waiting for five or more people who will be waiting to sign in their attendance then take my turn. It took me a while to come up with a draft solution then I decided to design a facial recognition based attendance system which automatically signs in your attendance as you arrive at your work place.

**REQUIREMENTS AND TECHNOLOGIES USED**

* This project requires the use of a camera for recognizing and capturing faces.
* I used the Python programming language and used the Viola-Jones Algorithm which uses a Haar cascade classifier to detect faces in an image or video. The algorithm is a popular technique in computer vision, works by scanning the image with a sliding window and analysing the pixel values of each window to determine if it contains a face.
* I also used the streamlit framework for the user interface and is interaction friendly.

**Steps for performing face detection using Cascade Classifier:**

1. Load the pre-trained classifier: A pre-trained classifier for face detection is provided with OpenCV. This classifier is stored as an XML file and can be loaded using the CascadeClassifier class.

*face cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml')*

1. Load the image: The image to be processed is loaded using the imread function.

*img = cv2.imread('image.jpg')*

1. Convert the image to grayscale: The CascadeClassifier algorithm requires the input image to be in grayscale format. The cvtColor function is used to convert the image to grayscale.

*gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)*

1. Detect faces: The detectMultiScale method of the CascadeClassifier class is used to detect faces in the grayscale image. This method returns a list of rectangles that contain the detected faces.

*faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5)*

* scaleFactor: will decrease the detection time but may miss smaller faces if increased.
* minNeighbors: will reduce false positives but may miss some faces.

1. Draw rectangles around detected faces: The rectangle function of the OpenCV library is used to draw a rectangle around each detected face.

*for (x, y, w, h) in faces:*

*cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)*

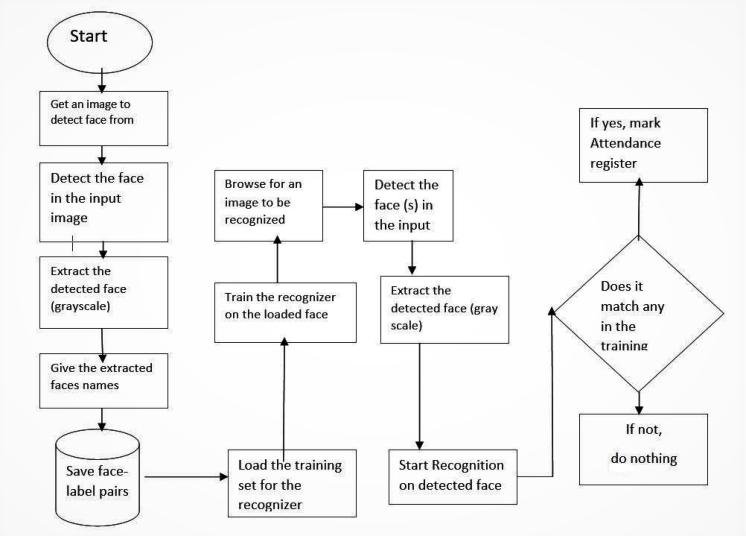
1. Display the image: The imshow function of the OpenCV library is used to display the image with the detected faces.

*cv2.imshow('img', img)*

*cv2.waitKey()*

The waitKey function is used to wait for a key press before closing the window.

**CLASS DIAGRAM**

****

**CONCLUSION**

Facial recognition and detection using CascadeClassifier, and different programing algorithms are simple and effective techniques for developing this project. By adjusting the parameters, the sensitivity and accuracy of the algorithm can be optimized for different use cases.