Great Place To Work® Certified MAR 2021-FEB 2022 INDIA

UPES UNIVERSITY OF TOMORROW

CLOUD APPLICATION DEVELOPMENT

Title:

Cloud Based Attendance System Using Facial Recognition in Python

Presented by:

Kumar Dev 500083164

Guided by:

Dr. Harvinder Singh

Assistant Professor

Department of Systemics

1. Introduction

- An innovative solution that automates attendance tracking for businesses is a cloud-based attendance system that uses Python facial recognition technology. It authenticates users using facial recognition technology to identify them and ensure that attendance records are correct in real-time. Because the system is housed on a cloud platform, it is scalable, dependable, and secure. The system provides capabilities including real-time tracking, attendance control, and system integration. It is a complete attendance management solution that reduces the need for manual tracking and offers more accuracy and efficiency.

2. Problem Statement

The Problem is to Organisations that track employees' attendance manually are more likely to make mistakes, find anomalies, and manipulate attendance data. Additionally time-consuming, this approach results in inefficiencies and lower productivity. By automating the attendance monitoring process and maintaining accurate records in real-time, a cloud-based attendance system employing Python and facial recognition can address these problems.

3.Project Proposal:

- **Objective:** The Objectives of a cloud-based attendance system using Python facial recognition are to automate and simplify attendance tracking, improve accuracy and eliminate errors, offer customization and anomaly detection, provide real-time tracking and reporting, increase productivity, and enhance workplace security.
- **Target Users**: Organisations and businesses of all sizes and in all sectors that need a reliable and effective means to track attendance are the target users for a cloud-based attendance system employing facial recognition in Python. This can include institutions like colleges, hospitals, government offices,

manufacturing sites, and more. Managers, supervisors, and HR staff can maintain attendance records and provide reports using the system. Employees may also utilise the system to check in and leave, making it a practical and simple way to keep track of attendance.

2. Understanding the relationship between different kinds of Threads

- There are two ways to execute code concurrently in a programme: threads and processes. A thread can operate concurrently with other threads within a process, whereas a process is a unit of execution that is governed by the operating system.
- By employing threads, a work can be broken down into smaller subtasks that can be carried out concurrently on many threads.

Utilising the platform's APIs and tools to build and manage threads within the application is a need for developing apps that use the cloud application platform's threads.

Multi-threading In Attendance System Using Facial Recognition

An application's performance can be increased by using the multi-threading technique, which enables it to carry out numerous activities at once. Python can leverage multi-threading to speed up the capture and processing of photos as well as the detection of faces in a cloud-based attendance system. By doing this, the amount of time needed to record everyone's attendance can be considerably decreased.

You can identify the sections of the code that can be run concurrently, such as the image processing and facial recognition operations, to implement multi-threading. The threading module in Python can then be used to generate several threads, each handling a distinct task. These threads can operate independently and concurrently, enhancing the application's overall performance.

In conclusion, you may speed up the process of recording attendance for a sizable number of individuals and enhance the overall performance of the programme by employing multi-threading in a cloud-based attendance system that uses Python and facial recognition.

Approach To Implement Multi Threading

- Recognise the issue and specifications.
- Create the architecture for the system.
- Pick a multi-threading library that works for you.
- Find out how many threads there are.
- Implement the code for multi-threading.
- Performance should be tested and improved.

UPES

UNIVERSITY OF TOMORROW

Thank You