ATTENDANCE MATE

Face Recognition Attendance System

- Hrithik Mhatre
- Mehul Agrawal
- Sanidhya Garg
- Pradyuman Agrawal

PROBLEM STATEMENT

The problem we are solving is the need for an efficient and accurate attendance system in our college. Traditional attendance systems can be time-consuming and prone to errors. By developing a facial recognition system, we can streamline the attendance process and reduce the likelihood of errors. This system can also provide real-time attendance tracking, allowing for better monitoring of student attendance and potentially improving student engagement. It can also be used to streamline everyday activities such mess attendance, Reading Hall attendance etc Overall, the implementation of a facial recognition attendance system has the potential to significantly impact the efficiency and accuracy of attendance tracking in our college.

PROJECT DESCRIPTION

Create a Python-based facial recognition attendance system that is highly customizable and capable of detecting and recognizing faces with exceptional accuracy. The system should be able to mark attendance from photos captured within our app. In addition to Python, the system will incorporate various machine learning algorithms and facial recognition libraries to achieve optimal results.We will utilize this to record attendance in various settings such as classrooms, reading halls, and mess areas.As this project relies solely on an app and does not necessitate any physical camera installation, **the cost of implementation of this idea is very minimal.**

PROJECT DEPLOYMENT

- We plan to deploy our project, the Python-based facial recognition attendance system, on campus to streamline the attendance and entry process to enhance efficiency.

- The system will be implemented in various settings such as classrooms, reading halls, hostels and mess areas.

- Benefits for the community and stakeholders:
  - Improved accuracy and reliability of attendance tracking
  - Time savings for both students and faculty
  - Real-time monitoring and reporting
  - Reduction in manual record-keeping errors
  - Insights for administrative decision-making

TOOLS

Software tools:

- Python 3, OpenCV, keras, tensorflow, Pandas and CSV libraries, NumPy
- Haar cascade frontal face detector, Mediapipe face detection library, face_recognition library, OS library, Datetime, , Vortex AI, Roboflow, YOLO V5
- PHP, Appium, Some IDE (Probably VS Code)

Hardware tools:

- Utilization of existing devices (e.g., smartphones with cameras) for capturing photos
- No additional physical camera installation required, resulting in minimal implementation costs
- Integration of the app with phone's camera for seamless attendance recording

Contribution and Learning

- **Sanidhya**: He has already done a similar project under a professor in IITKGP so he has a good idea on what softwares should be used and how should we go about implementing the project. He will contribute in writing the code for the model and training
- **Mehul**: He has experience with making webpages, both the frontend and backend. So he will be developing the app. Plus he also has worked with ML so he will also contribute in training the model
- **Hrithik**: Having worked with cameras and interfacing before, he will take care of integrating the model with the app along with integrating the app with phone's camera
- **Pradyuman**: He has experience in the field of analytics so he will be preparing the dataset of images for our model and handle the pre-processing.

Learning so far:

- Discover more about deep learning and convolutional neural networks
- Learn how to create a machine learning model using Python libraries like Tensorflow, keras and pytorch
- A fundamental understanding of hyper parameters, transfer learning, and fine tuning

Learning opportunities:

- Deeper understanding of computer vision through research papers and practical knowledge
- Data augmentation, app development, model integration and deployment
- Getting well verse with software like - python3, YOLO v5, Roboflow, Vertex AI

Timeline

Tentative timeline[After taking into consideration the future academic commitments and POR]:
May (some would be doing intern and some would be at home so not much progress can be done):
• Gaining deeper understanding of CNN and siamese network • Gathering resources(Vertex AI, Roboflow)
June (same as above):
• Dataset preparation • Code for Face detection using roboflow and yolo with active learning
July:
• Code for Face recognition (debugging, testing for single face detection) • Training the ML model (debugging)
August-September (midsems time) :
• APP development
October-December:
• Integration of ML model

Budget

As mentioned before, this project relies solely on an app and does not necessitate any physical camera installation, the cost of implementation of this idea is very minimal.The only expense that would be required which we can see as of now if for Google's vertex AI which has separate pricing for training, deploying and using and would amount to about $26/hr or ₹2100/hr when in use although we can't give an exact estimate for how much time will we use it as of now.

Work Done till Now

- Completion of deep learning specialisation course by Andrew Ng, which includes-
  - Understanding of Neural Networks and deep learning
  - Hypertuning, optimization and regularization
  - Convolutional Neural Network
  - Transfer learning
- Going through some computer vision research papers:
  - Zeiler and Fergus., 2013, Visualizing and understanding convolutional networks
  - Taigman et. al., 2014. DeepFace closing the gap to human level performance
  - Schroff et al., 2015, FaceNet: A unified embedding for face recognition and clustering