MINeD Hackathon Report

Team: ‘ERROR 404’

Team Members:
Aditya Singh
Akash Arya
Aryan Mehta
Kanav Avasthi
Avi Tayal

Track:
JK Lakshmi Cement: Face Detection using Computer Vision

Introduction

Face detection technology has received a lot of attention recently and has been used in a variety of industries, including security, surveillance, and marketing. Employee face detection has become a viable technique for increasing security, productivity, and efficiency in the workplace. Advanced computer vision techniques can precisely identify and track employee faces in real-time, giving important details about their conduct, attendance, and engagement. We want to design a system that can improve workplace administration and operations while also investigating the potential of employee face detection in this project.

Model

Dataset Used:

We are provided with 2 folders. There is an image folder containing employee images to be used for comparison, there is another folder containing a video file which has to be used t

Libraries Used:

1. Numpy: Python's NumPy library is used to perform numerical computations. Together with a range of functions for performing mathematical operations on those arrays, it offers strong data structures for the effective handling of big arrays and matrices. In scientific computing, data analysis, and machine learning, NumPy is frequently used.

2. Os: The Python standard library's os module offers a means of interacting with the operating system. It offers a number of tools for interacting with processes, environment variables, the file system, and other things.

3. Face_recognition: A Python package called the face_recognition library gives programmers the ability to identify and work with faces. Facial identification, face detection, face landmarking, and face encoding are all performed using a combination of deep learning and computer vision methods.

4. Cv2: A free computer vision library for Python is the cv2 library. For processing, analysing, and altering pictures and videos, it offers a selection of tools and algorithms. The OpenCV (Open Source Computer Vision) library, a popular computer vision library is the foundation around which the library is constructed.

5. Pandas: Pandas is an open-source library for data manipulation and analysis. It provides data structures and tools for working with structured data, including tabular data, time-series data, and matrix data.

Process:

The steps for creating a facial recognition model in Python are listed below.

1. Import necessary libraries
   Libraries: cv2,os,face_recognition,numpy,pandas

2. Load and prepare training data:
   The first step in training the face recognition model is to set a path for our image folder. We then import the directory provided to us containing employee images. The image directory path is stored for further use.

3. Image Encoding:
   We run a loop across every image in our directory to load the images and encode them. We store both of them in separate lists.

4. Output Video Location:

We define the path where our processed video could be stored. We also define an fps equal to that of the input video so that they could run at a similar speed.

5. Import the video:
   The provided video is imported and we run a loop on it to extract every frame as long as the video runs. NOTE: We do propose a model which only takes into consideration every 10th frame rather than using every single frame for processing as it could reduce processing time significantly.

6. Convert images into BGR format and detecting faces
   We convert our frames from RGB to BGR form. We use face encoding and detection parameters of the face_recognition module to locate the coordinates of detected faces in a frame and draw a rectangle around it.

7. Comparing Detected Faces to our dataset:

   We use a distance hyperparameter of 0.515 to compare detected faces to the faces in the dataset. We display the name of the detected faces under the rectangle.

8. Declaring Unknown Faces:

   If the distance hyperparameter is detected at less than 0.515, we declare it as unknown.

9. Converting back to RGB for video transmission:
   In order to display our video we transform our BGR frames back to RGB format. We then release the video.

Data Storage:

Faces are detected and stored in a list based on the earliest frame of appearance, the time of occurrence is detected from there. The employees in the video are registered in a list which is converted into a .csv file along with the timestamp of their appearance.