FACIAL RECOGNITION SYSTEM DEVELOPMENT DOCUMENTATION

By Group 2

System Overview

The Facial Recognition System is a system developed using Python programming language with the aim of identifying and verifying people based on machine learning techniques and other methods learnt from the course Artificial Intelligence (CSC309). This system will use a dataset consisting of a number of labeled images to match the faces detected from other images against the dataset. It would return the identity of the face if it matches any of the records in the dataset; otherwise, it will return "Unknown".

Methodology

Developing a Facial Recognition System using Machine Learning Techniques involves a set of systematic steps.

- Problem Definition

The objective of the facial recognition system is to identify and verify a person from an image based on a defined dataset. Some issues such as occlusion, varying lighting conditions and pose variations are challenges that will be faced when developing this system.

- Data Collection

For the purpose of developing this system, a dataset of images used to train the model was created. The dataset consists of images manually uploaded and labeled, and can be updated with images taken by the webcam from this system in real time. The dataset is rather small as this is a small project, though this limits the accuracy of the system to some extent.

- Data Preprocessing

Raw images collected from the webcam and those manually uploaded need to be prepared for analysis. For those collected from the webcam the algorithm 'Haar Cascades' was used to detect faces in images, and capture precisely those faces.

- Feature Extraction Using Deep Learning Methods

  The Python module ‘face_recognition’ leverages the pre-trained model, ResNet-34, to extract high-dimensional embeddings and create face encoding for the detected faces in the images. The module was then fine-tuned on the dataset created for an efficient performance.

- System Evaluation

  The system was then evaluated using the metrics below:

  - **Accuracy**: The measure of how often the system correctly recognizes a known face and an unknown.
  - **Precision**: The measure of how often the system is correct when it recognizes a known face.
  - **Recall**: The measure of how often the system recognizes known faces.
  - **F1-score**: The balanced measure of the system’s precision and recall.

Setup and Usage

This system was built in the Python Version 3.11.9 environment. It can run efficiently in the absence of a GPU without overworking or overloading the computer system. The Python Library, OpenCV (Open Source Computer Vision Library), was installed and used in the live system dataset collection process. The modules: OS, NumPy, Face_Recognition were installed and used in the development of this system.

To run and use this system on any computer simply go through the following steps:

- Download and unzip the zip file, Face Recognition System Group 2.zip.
- Ensure you have pip installed on your computer, then open terminal in the folder and run:
  - pip install os
  - pip install numpy
  - pip install opencv-python
  - pip install face_recognition
- Run the ‘main.py’ source code and the system should be up and running (password for Admin is ADMIN).
- OPTIONAL
  - Connect a webcam to your computer (**recommended**) - live face recognition and data collection cannot be done without it.
  - Change the test images in the folder to any image of your choice, if the names of your choice images are different remember to change the image path in the source codes.

- Open the dataset folder, images, and edit the out-of-box dataset to images of your choice with clear faces of people in them. Name the files with the labels of the image (Eg: John.jpg).

Dataset and Models

In the course of developing this system, a dataset was created consisting of a few pictures of people labeled with the name of the person - the image name is used as the name of the person. The dataset allowed for images to be manually uploaded and labeled or taken live and labeled from the system by an 'Admin'.

The 'Haar Cascade Classifier' was used to detect faces from the webcam and to take precise and focused images of each person for populating the dataset. The Haar cascades (*XML files*) are the basis for OpenCV's method of face detection. They identify and consist of certain patterns or characteristics in pictures that correspond to facial features like the eyes, nose, and mouth in the context of face detection. This dataset was then used by the trained model in order for the system to process images, detect faces and match faces to people. The dataset is stored in a folder titled 'images'.

The Python module 'face_recognition' was used in building this system. It provides a simple user interface for face detection and recognition tasks. It makes use of the deep learning-based face identification algorithm from Dlib, which in turn uses face encodings or embeddings to recognize faces. Face encoding is a numerical representation of the face that captures the key facial traits of a person's face like distance between eyes, nose and mouth, distance between the forehead and chin, etc. Dlib applies the ResNet-34 deep learning model to generate these face encodings. Finally, the system assesses the likeness between the encodings, or matches faces against the array of recognized faces.

Testing and Results

This system was tested with images to verify known faces, and with live faces captured from the WebCam. It was tested in different environments and under different conditions, such as lighting and capture angles. An average of 65 cases where tests, and the result of the tests are as follows:

True Positives = 25 True Negatives = 25
False Positives = 9 False Negatives = 6
The accuracy was determined to be 0.769 = 76.9%
The precision was calculated as 0.735 = 73.5%
The recall was determined to be 0.806 = 80.6%
The f1-score was calculated as 0.768 = 76.8%

In summary, the system is more likely to determine an unknown face as a known face, than to determine a known face as an unknown face. Therefore, it is of better use in situations and large systems where it will be less harmful in low risk scenarios like a Customer Engagement System that uses facial recognition for personalized services, offers and greetings; If an unknown face is incorrectly classified as a known face, the result might simply be a friendly greeting or an offer, which poses no major harm. However, missing a known face (a valued customer) could result in a loss of engagement or a bad customer experience.

Group Members

1. Chizim Awele Gabriela - 22120612972
2. Captain-Briggs Gina Oluwatamilore - 21120612631
3. Charles-Agu Jesse Agu - 22120612970
4. Daniel Jeremiah Chijoke - 22120612973
5. Egwuatu Ifunaya Nicole- 22120612976
6. Femi-Sunmonu Abdulrazaq Olamidotun - 22120612985

Tasks

1. Dataset Collection and Collation - (Jeremiah & Jesse)
2. Face Detection - (Abdulrazaq & Awele)
3. Image Encoding and Labeling - (Abdulrazaq & Awele)
4. System Testing - (Jeremiah & Jesse)
5. Documentation and Research - (Gina, Ifunanya & Awele)