CSE 429

Computer Vision

Term Project: Automated Attendance System Using Face Recognition and Deep Learning

(3/01/2023)

Submitted by

Ashrakat Saeed - 120200091

Abdelrahman Said - 120200075

Mostafa Atef - 120200043

Mariam Ayman - 120200094

Asem Mohamed - 120200089

Group 1 – Section 1

TO

Dr/Ahmed Saleh & Eng. Mahmoud Sharshira

Table of Contents

I. Project Overview

The Face Recognition System is a computer vision-based project designed to recognize and identify individuals based on their facial features. The system utilizes state-of-the-art deep learning and image processing techniques to achieve accurate and real-time face recognition. Additionally, a Flask web application has been integrated to facilitate attendance tracking through a Google Sheets spreadsheet.

II. Key Features

- **Face Detection**: Utilizes Dlib's face detector to locate faces in input images.
- **Face Alignment**: Uses facial landmarks to align detected faces for consistent recognition.
- **Face Encoding**: Computes 128-dimensional facial encodings using a pre-trained ResNet model.
- **Real-time Recognition**: Provides real-time face recognition and identification from images.
- **Attendance Logging**: Logs recognized individuals and timestamps in a Google Sheets spreadsheet via a Flask web application.

III. Dependencies

Dependency	Function
gspread	Python API for Google Sheets.
OpenCV	Open-Source Computer Vision library for image processing.
Dlib	C++ toolkit containing machine learning algorithms and tools.
NumPy	Fundamental package for scientific computing with Python.
face_recognition	Face recognition library for Python.
dlib shape_predictor and face_recognition_model_v1 files	Pre-trained models for facial landmark prediction and face encoding.
Flask	Micro web framework for building web applications.

IV. Data Collection & Preprocessing

1- Data collection

- Existing Dataset

The initial dataset comprised 53 videos, each lasting 5 seconds, for each individual. However, a significant challenge arose due to the lack of diversity within the dataset. The limitation was particularly evident in scenarios where only a single video existed for each person, and the frames within these videos exhibited high similarity.

- Diversification Efforts

Recognizing the importance of diversity for robust face recognition, additional measures were taken to augment the dataset. These efforts included:

1- Social Media Platforms:

Facebook and LinkedIn: Data collection extended to social media platforms, where profile photos from Facebook and LinkedIn were acquired. This allowed for a broader representation of individuals in different settings.

2- Personal Interaction:

IN cases where online profiles were insufficient, personal interaction played a crucial role. Colleagues were approached individually, and consent was obtained to capture facial images for the dataset. This direct engagement helped enhance the dataset's inclusivity.

The combination of the original video dataset and the additional images obtained from various sources significantly contributed to achieving a more diverse and representative dataset. This diversity is fundamental for training a face recognition model capable of handling a wide range of scenarios, lighting conditions, and facial expressions.

2- <u>Preprocessing</u>

Before training the model, preprocessing steps were applied to ensure data uniformity and enhance model performance. These steps included:

1- Face Detection and Alignment:

Leveraged Dlib's face detector and facial landmarks predictor to detect faces and align them consistently across all images.

2- Image Cropping:

Cropped images to focus specifically on facial regions, eliminating unnecessary background noise.

3- Data Labeling:

Labeled data with corresponding identities to facilitate supervised learning.

V. Modeling

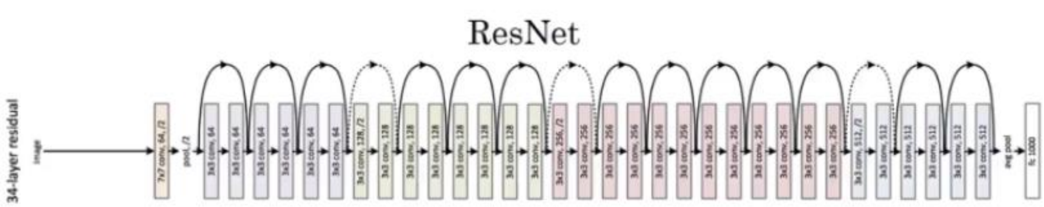
1- Model architecture

The face recognition model employed in this project is a variant of the ResNet architecture, featuring 29 convolutional layers. Derived from the ResNet-34 network presented in the paper "Deep Residual Learning for Image Recognition" by He, Zhang, Ren, and Sun, this adapted version has undergone modifications. Notably, several layers were removed, and the number of filters per layer was halved to optimize performance.

Training Details

Training Dataset: The model was trained from scratch on a diverse dataset comprising approximately 3 million facial images. This dataset is a compilation from various sources, including:

- Face Scrub Dataset
- VGG Dataset



ResNet-34

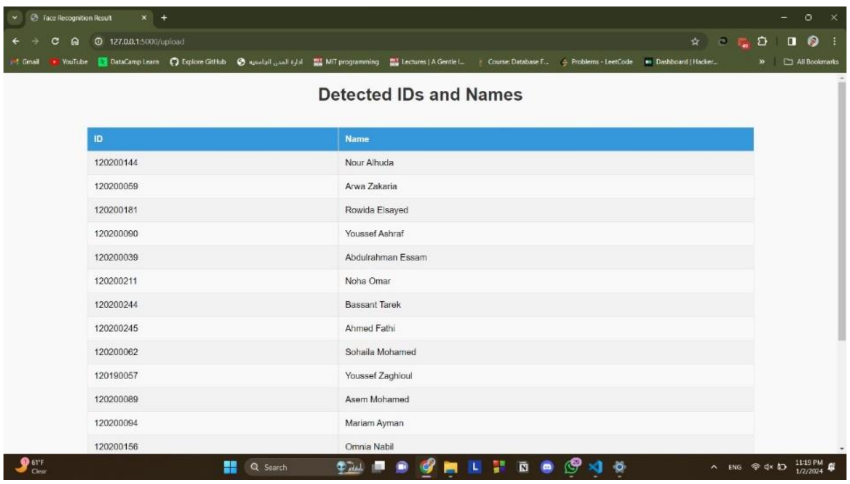
2- Model training

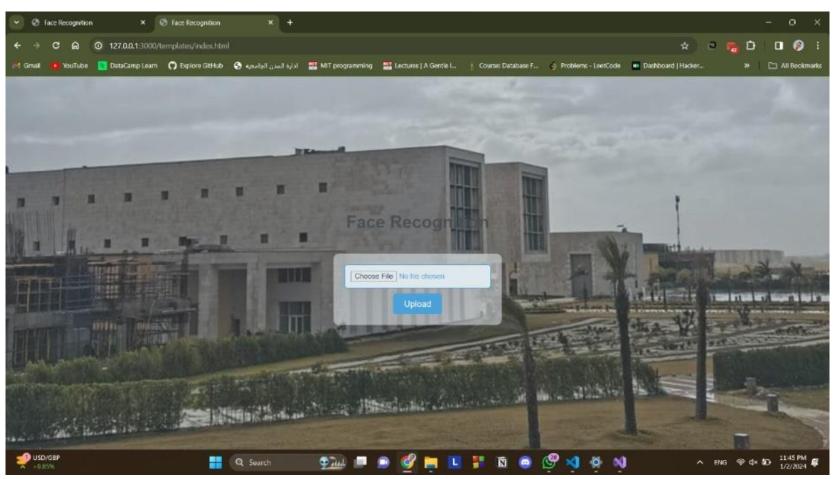
- Utilized a pretrained ResNet model with 29 convolutional layers initially trained on a diverse dataset of 3 million faces.
- Fine-tuned the model on a custom dataset featuring 53 individuals and 260 images to address limited diversity.
- Augmented the dataset by collecting additional facial images from diverse sources like Facebook, LinkedIn, and personal requests.

- Employed transfer learning techniques to adapt the pretrained model to the specific facial characteristics of the custom dataset.
- The approach showcases the model's adaptability to unique facial characteristics and its effectiveness in facial recognition tasks as **it recognizes 21 faces on test image from 22 in its best performance, and it detects them all.**

VI. Web Application

The Flask web application allows users to upload images for face recognition. The recognized individuals and timestamps are automatically logged in a Google Sheets spreadsheet. The Flask app provides a simple and user-friendly interface for attendance tracking.





VII. Conclusion

In conclusion, our project has realized an automated attendance system through the integration of computer vision and deep learning. This efficient and user-friendly solution streamlines the attendance tracking process, offering a reliable tool for time-saving and effective management.