Image Processing – Attendance System

The Attendance Sheet Management System aims to streamline and digitize the process of recording attendance for organizations, classes, or events. This project involves the development of an efficient and user-friendly digital platform/program that allows for the easy and accurate tracking of attendance. By leveraging powerful computer vision and machine learning tools, this system can revolutionize the way attendance is managed, providing organizations with a more accurate, reliable, and convenient solution.

by Ahmed Algrgawy

Project Dependencies

1 OpenCV

OpenCV, or Open Source Computer Vision Library, is a powerful open-source computer vision and machine learning software library primarily developed in C++, with interfaces for Python and other languages. It's widely used for tasks such as image and video analysis, object detection and recognition, face detection, and more.

2 cv2 Module

The cv2 module in Python is the interface to the OpenCV library. It provides functions and classes to perform various image processing and computer vision tasks. cv2 is often used alongside other Python libraries like NumPy for efficient array manipulation and Matplotlib for visualization.

3 OS Module

The os module in Python provides a way to interact with the operating system. It allows you to perform various tasks such as navigating file systems, manipulating files and directories, accessing environment variables, and executing system commands.

4 DeepFace

DeepFace is a deep learning-based face recognition system developed by Facebook AI Research (FAIR). It's designed to perform accurate face verification and recognition tasks using deep neural networks.

5 RetinaFace

RetinaFace is a state-of-the-art face detection and alignment algorithm that accurately locates and aligns faces in images with high precision. Developed by researchers from the Chinese University of Hong Kong and Alibaba Group, RetinaFace employs a single deep neural network architecture to simultaneously perform face detection and facial landmark localization.

Additional Dependencies

Matplotlib.pyplot

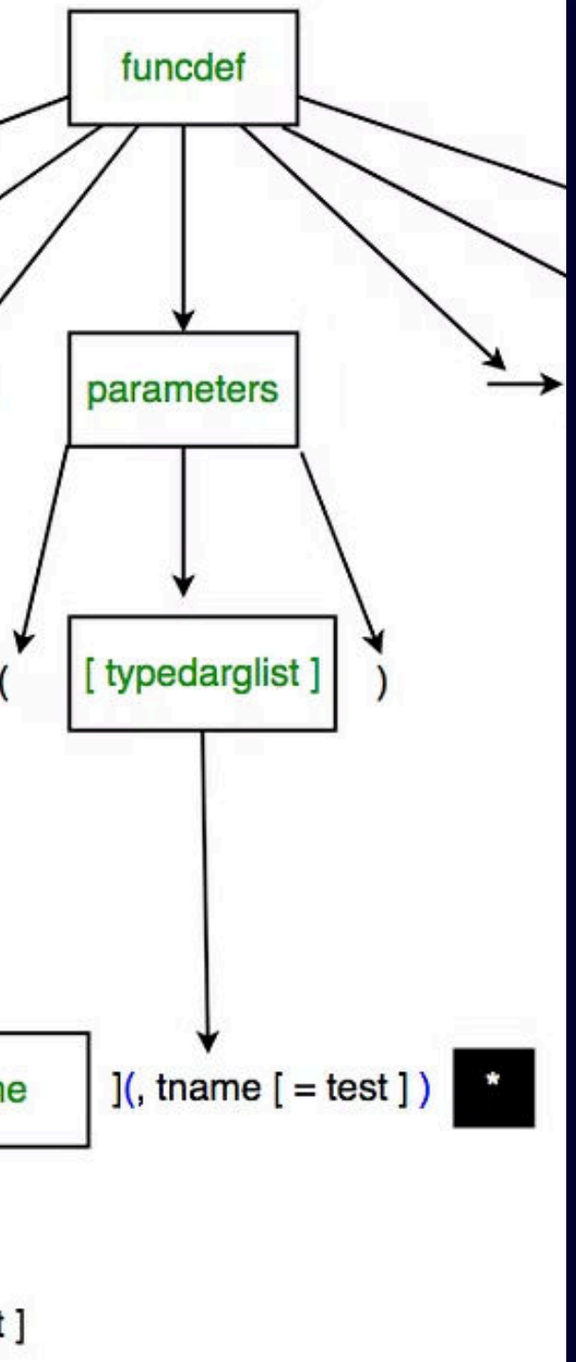
Matplotlib.pyplot is a Python library used for creating static, animated, and interactive visualizations in Python. It is part of the Matplotlib library, which is one of the most widely used plotting libraries in the Python ecosystem.

Pandas

Pandas is a powerful open-source Python library used for data manipulation and analysis. It provides easy-to-use data structures and functions for working with structured data, primarily in the form of tabular data (like spreadsheets or databases).

Tkinter

Tkinter is a Python library used for creating graphical user interfaces (GUIs). It's a standard library included with most Python installations, making it a convenient and accessible choice for building desktop applications.



Utility Functions

1

getVideoPaths(folderPath)

This function retrieves paths of video files (with specific extensions) from a given folder. It iterates through files in the specified folder, checks if the file ends with one of the specified video extensions (.mp4, .avi, .mkv, .wmv), and appends the path of the video file to a list.

2

extractFrames(videoPath, outputFolder)

This function takes in a video file path, an output folder path, and two optional parameters: check_existing and fbs. It extracts frames from the video file and saves them into the specified output folder. It also handles checking for existing frames and determining the frames per second (FPS) of the video.

3

get_photos(folderPath)

This function takes in a folder path and returns a list of paths to image files (photos) found in the folder. It iterates through files in the specified folder, checks if the file name ends with one of the specified image extensions, and appends the path of the image file to a list.

Frame Extraction Process

1

Specify Videos Folder

The script defines the path to the folder containing input video files (videosFolder).

2

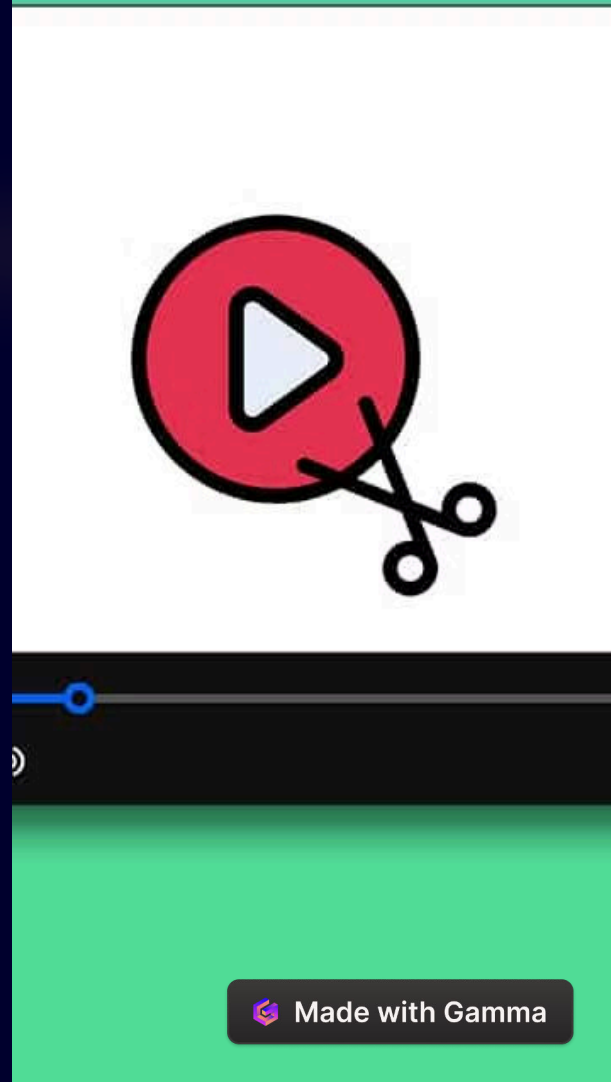
Retrieve Video Paths

The script calls the getVideoPaths() function to retrieve paths of video files from the specified folder (videosFolder).

3

Extract Frames

For each video file, the script constructs the output folder path and calls the extractFrames() function to extract frames from the video and save them into the specified output folder.



Face Detection and Recognition

Retrieve Photo Paths

The script calls the get_photos() function to retrieve paths of photo files from the specified folder (photosFolder).

Face Detection

For each photo, the script extracts faces using the RetinaFace algorithm and prints the number of detected faces.

Face Recognition

The script attempts to recognize the detected faces using the DeepFace library, displaying the recognized faces with their assigned numbers or indicating that a face was not recognized.

Data Handling

The script saves the recognized face numbers to a pandas DataFrame and exports the data to an Excel file named "Attendance.xlsx".

Attendance Data Visualization

Data Visualization

The script utilizes Matplotlib.pyplot to display the detected and recognized faces, providing a visual representation of the attendance data.

Excel Integration

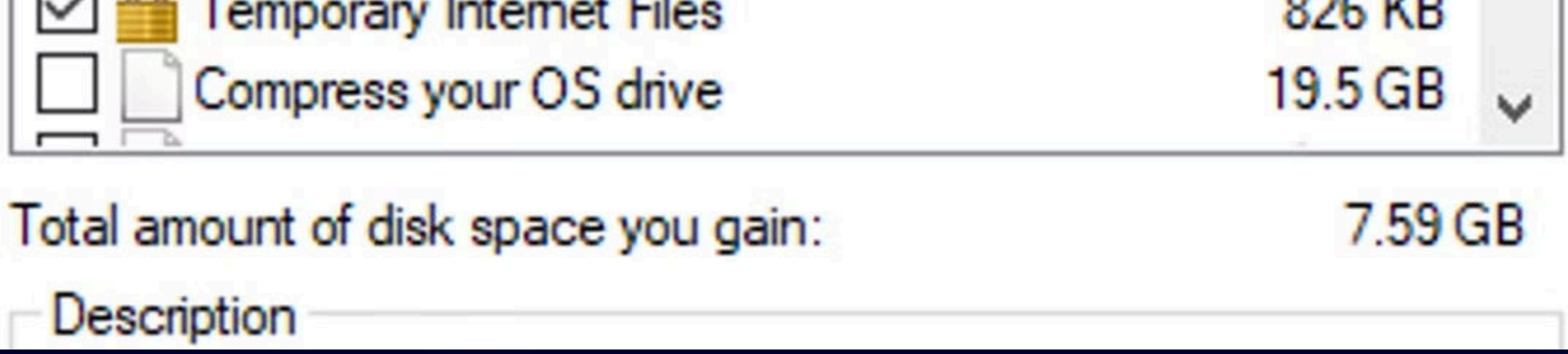
The attendance data is saved to an Excel file, allowing for easy integration with existing attendance management systems and further analysis.

Automation

The script's ability to automatically extract frames, detect faces, and recognize individuals streamlines the attendance tracking process, reducing manual effort and improving efficiency.

Scalability

The modular design and use of powerful computer vision and machine learning libraries enable the system to handle large-scale attendance data and adapt to the needs of various organizations.



Temporary File Cleanup

1 Removing Temporary Files

The script cleans up the temporary file "detectedFace.jpg" that was used during the face detection and recognition process, ensuring a tidy and organized file system.

2 Maintaining Efficiency

By removing temporary files, the script helps maintain the system's efficiency and prevents the accumulation of unnecessary data, which can impact performance and storage requirements.

3 Best Practices

The cleanup process aligns with best practices for file management, promoting a well-organized and maintainable codebase that adheres to principles of clean and sustainable software development.

Attendance Data Management

Pandas DataFrame

The script utilizes the Pandas library to create a DataFrame named "AttendanceData" that stores the recognized face numbers. This structured data format allows for efficient data manipulation and analysis.

Excel Integration

The attendance data is saved to an Excel file named "Attendance.xlsx" using the Pandas to_excel() function. This integration with a widely-used spreadsheet application facilitates easy access, sharing, and further processing of the attendance records.

Data Preservation

By saving the attendance data to an Excel file, the script ensures the long-term preservation of the attendance records, enabling organizations to maintain historical data and track attendance trends over time.

Conclusion

The Attendance Sheet Management System showcases the power of computer vision and machine learning technologies in streamlining and digitizing the attendance tracking process. By leveraging OpenCV, DeepFace, RetinaFace, and other Python libraries, the system can accurately detect and recognize faces, extract attendance data, and store it in a structured format for easy access and analysis. The integration with Excel further enhances the system's usability and compatibility with existing attendance management workflows. This comprehensive solution not only improves the efficiency of attendance tracking but also provides organizations with valuable insights and historical data to support their decision-making processes. The modular design and use of cutting-edge algorithms ensure the system's scalability and adaptability to the evolving needs of diverse organizations.