ATTENDANCE SYSTEM USING FACE RECOGNITION

-PIYUSH BHATNAGAR

**READ THE views.py FILE. IT CONTAINS ALL THE FUNCTIONS !!!!**

Models:

There are two models: user, present, time

* User: default model used for users login
* Present: (tells whether a particular user is present on a particular date or not)

->has user which is foreign key for User model (if a user gets deleted then all its p resent records will also be deleted)

->date: default date as today

->present: default False

* Time: (tells the time of in and out of a particular user on a particular date)

->user (foreigne key), {Time rows for that user gets deleted when user gets deleted}

->date

->time

->out (true or false)

Database used: postgresql

Q. Explain your project?

This is an face recognition attendance system. Employees can mark their attendance on this website and system will identify the user and record it's attendance.

Instead of traditional methods like signing a paper or swiping a card, this system scans the faces of indivisuals. Admin can view attendance records of each employee.

Employees can also view his attendance records.

Q. Short explanation of my face recognition attendance system?

Your face recognition attendance system is a modern technology that uses facial recognition algorithms to accurately identify and record the attendance of individuals. Instead of traditional methods like signing a paper or swiping a card, this system scans the faces of individuals as they enter a designated area. It matches their facial features against a database of known faces, allowing for quick and efficient attendance tracking without the need for physical interaction. This system offers convenience, security, and accuracy in managing attendance records.

Q. But what is the need of this system for attendance like what problem it solves?

Accuracy: Manual entry or card swiping can be prone to errors such as incorrect entries or one person signing in for another. Face recognition technology ensures accurate attendance tracking by uniquely identifying individuals based on their facial features.

Efficiency: With face recognition, the process of recording attendance becomes faster and more efficient. There's no need for individuals to carry or swipe cards, or for staff to manually record attendance data. This saves time and reduces administrative burden.

Security: Face recognition adds an extra layer of security to the attendance tracking process. It helps prevent unauthorized access by ensuring that only registered individuals can gain entry, reducing the risk of impersonation or unauthorized attendance.

Contactless Interaction: Particularly relevant in situations where hygiene is important (such as during a pandemic), face recognition allows for contactless attendance tracking. There's no need for individuals to touch shared surfaces like fingerprint scanners or keypads, minimizing the risk of spreading germs.

Q. Is there any drawback for this face recognition attendance system?

Privacy Concerns: Facial recognition technology involves capturing and storing biometric data. Some people may feel uncomfortable with the idea of their facial data being collected and stored by an organization.

Accuracy Issues: Changes in lighting conditions, facial expressions, or the presence of accessories (like glasses or hats) can sometimes affect the accuracy of recognition. This can lead to instances of misidentification or false positives, which may result in attendance recording errors.

Cost: High cost of hardware like high quality cameras and software.

Q. What if someone marks other attendance using his photograph?

Accuracy of Recognition: High-quality face recognition systems are designed to accurately distinguish between real faces and photographs. Advanced algorithms analyze various facial features and characteristics to verify the authenticity of the presented face.

User Authentication: Some face recognition systems require additional forms of authentication, such as a PIN or biometric verification (e.g., fingerprint scan), to supplement facial recognition. This adds an extra layer of security and makes it more difficult for someone to fraudulently mark another person's attendance using a photograph.

Monitoring and Oversight: Organizations may implement monitoring mechanisms to detect and deter fraudulent behavior. This could include reviewing attendance records for anomalies, conducting periodic audits, or implementing surveillance cameras to monitor the attendance marking process.

#dlib's shape\_predictor works only with HOG's face detector which we can use using dlib.get\_frontal\_face\_detector

#'detector' will detect the faces in the given grayscale image and will give us the coordinate where the faces are located

#'predictor' will give the coordinates of all the 68 face landmarks in the face in the image

STEPS :-

->We capture the image and convert it to grayscale

->then we detect face using detector then we get the coordinates of the faces in the image then we align the image

->they we get the details of the 68 landmarks of face using predictor

Q. How you create dataset?

We use dlib's face detector and shape\_predictor to create dataset. We capture user's image then convert it to grayscale because different images have different lighting condition, so it is better to normalize the images into grayscale then we detect face in it, we draw a rectangle around the face and get it's coordinates then this information goes to shape predictor then shape predictor marks 68 landmarks on the face and stores their coordinates information in the form of data.

->in the create\_dataset function we are only converting the image to grayscale because it is then easy to detect face in the image and then we crop the original image such that only face is visible.

We are making an SVC model to classify images :- IMP

<https://www.geeksforgeeks.org/support-vector-machine-algorithm/>

<https://www.geeksforgeeks.org/creating-linear-kernel-svm-in-python/>

Accuracy of the SVC MODEL :

* Accuracy: 0.9666666666666667

WORKFLOW :

* Admin register employee id.
* Admin create dataset for the employee.
* Admin can train the model for the dataset created.
* Admin can view the attendance reports acc. to date, employee id.
* Employee can see his attendance report.

QUESTIONS :

Q. How face recognition works?

Face recognition works by capturing an image or video frame containing one or more faces, extracting features from those faces, and comparing them against a database of known faces. This comparison is typically done using algorithms that measure the similarity between the extracted features and the features of known faces.

Q. How detector and predictor works?

The detector identifies the location of faces within an image or video frame, while the predictor identifies facial landmarks such as eyes, nose, and mouth. These landmarks serve as reference points for feature extraction and face alignment.

Q. what is 68 landmarks predictor doing?

The 68 landmarks predictor identifies 68 specific points on a face, including corners of the eyes, nose, mouth, and chin. These landmarks are used to accurately align faces and extract features for recognition.

Q. How you access camera and save images?

In Python, you can access the camera using libraries like OpenCV (**cv2**). You can capture images using functions like **cv2.VideoCapture()** and save them using **cv2.imwrite()**.

Q. How to convert 68 landmarks values in dataset?

You can convert the 68 landmarks values into a dataset by storing them as feature vectors along with labels representing the identities of the individuals in the images.

Q. What are x,y in your dataset?

* X is the details of the face of person y.

Q. What are the labels in the dataset?

The labels in the dataset represent the identities of the individuals whose faces are being recognized. Examples include names like "Piyush\_0902" and "Abhinav\_07".

Q. Is face recognition system feasible?

Yes, face recognition systems are feasible and widely used for various applications, including security, authentication, and attendance tracking.

Q. Is there any limitation of this system?

Face recognition systems may face limitations in accuracy under challenging conditions such as variations in lighting, pose, expression, and occlusion. They may also raise privacy concerns related to data collection and storage.

Q. What problems you faced while making this project?

* Collecting the dataset then extracting faces and then facial landmark details from grayscale image is the main challenging work.
* Maintaining proper lighting in the images.
* I limited the no. of images to 100 for each user otherwise for traning it takes a lot of time.

Q. Why used Django? Not any other framework?

Django was chosen for its robustness, scalability, and built-in features for web development, including user authentication, database management, and URL routing. Its MVC architecture and extensive documentation also make it suitable for complex projects like this.

Q. Why used SQL as database?

SQL databases like SQLite or PostgreSQL are commonly used in Django projects due to their reliability, transaction support, and compatibility with Django's ORM (Object-Relational Mapping) system. They provide a structured way to organize and query data, essential for managing user information and attendance records in this system.

Q. Can we use NoSQL with Django?

Yes, you can use NoSQL databases with Django, although it may require some additional configuration. Django primarily supports relational databases like PostgreSQL, MySQL, and SQLite out of the box through its Object-Relational Mapping (ORM) system. However, you can integrate NoSQL databases like MongoDB or Redis with Django using third-party libraries or custom database backends.

Q. What is SVC model? When to use it? What is the use of linear kernel?

An SVC (Support Vector Classifier) model is a type of supervised machine learning model used for classification tasks. It belongs to the family of Support Vector Machines (SVMs), which are widely used for both classification and regression tasks.

Here's why SVC models are used and what the use of a linear kernel is:

1. **Why we use SVC models?** SVC models are popular for classification tasks because they are effective in handling both linearly separable and non-linearly separable datasets. They work by finding the optimal hyperplane that best separates the different classes in the feature space, maximizing the margin between classes while minimizing classification errors.

Key reasons for using SVC models include:

* + High accuracy: SVC models often achieve high accuracy in classifying data points into different categories.
  + Flexibility: They can handle datasets with complex decision boundaries and are less prone to overfitting compared to some other classifiers.
  + Versatility: SVC models can be adapted to various types of classification problems, including binary and multiclass classification.

1. **What is the use of a linear kernel in SVC?** In SVC models, the choice of kernel determines how the decision boundary is constructed in the feature space. A linear kernel is one of the kernel functions used in SVC, and it defines the decision boundary as a hyperplane in the original feature space.

The use of a linear kernel is beneficial in situations where the data is linearly separable, meaning the classes can be separated by a straight line or hyperplane. Linear kernels are computationally efficient and can handle large-scale datasets with high dimensionality.

Additionally, linear kernels are interpretable, making it easier to understand and interpret the decision boundary and the influence of different features on classification outcomes.

In summary, SVC models are used for classification tasks due to their effectiveness in handling various types of datasets, and the use of a linear kernel is particularly suitable when the data is linearly separable, offering computational efficiency and interpretability.