**CODE SUMMARY**

1. **IMPORTS**:

* cv2: OpenCV library for image processing.
* numpy: For numerical computations.
* face\_recognition: A library for facial recognition.
* os: For file and directory management.
* datetime: To handle date and time.

2. **DIRECTORY PATH**:

* The **path** variable points to a directory that contains images of individuals used for facial recognition.

3. **LOAD IMAGES**:

* The code reads all images from the specified directory.
* Each image's filename (without extension) is stored in **classNames** to represent the person's name.

4. **FUNCTIONS:**

* findEncodings(images): This function encodes the faces in the loaded images. It converts images to RGB format and extracts face encodings.
  + - Input: List of images.
    - Output: List of encodings.
* markAttendance(name): This function writes the detected person's name and current time to **Attendance.csv** if the person hasn't already been marked.
  + - Input: Person's name.

5. **INITIALIZATION**:

* Encodings for all images in the directory are calculated and stored in `encodeListKnown`.

**6. REAL-TIME FACE RECOGNITION**:

* A webcam feed is captured using `cv2.VideoCapture`.
* Each frame is resized and converted to RGB.
* Faces in the frame are located and encoded.
* Encodings from the webcam feed are compared to the known encodings using **face\_recognition.compare\_faces**.
* The face with the smallest distance is identified as the closest match.
* If a match is found, the person's name is displayed and their attendance is marked.

**ALGORITHM SUMMARY**

1. **PREPROCESSING**:

* Load and encode reference images of individuals.
* Store each person's name and face encoding.

2**. FACE DETECTION & RECOGNITION**:

* Capture real-time video from a webcam.
* Resize and convert each frame for processing efficiency.
* Locate faces and compute face encodings in each frame.
* Compare the current encodings with known encodings.
* Identify the person with the smallest distance match.

3. **ATTENDANCE LOGGING**:

* If the detected person has not been recorded, their name and current timestamp are logged in **Attendance.csv**.

**KEY ALGORITHMS**

* Face Encoding: Utilizes the **face\_recognition.face\_encodings()** function to generate a unique encoding (a vector of numerical values) that represents each face.
* Face Matching: The **face\_recognition.compare\_faces()** function checks if the current frame's encoding matches any known encoding.
* Distance Calculation: The **face\_recognition.face\_distance()** function measures the distance between the current encoding and each known encoding to determine the closest match.