**Face Recognition Project - Step-by-Step Breakdown**

**1. Adding Images for Recognition**

* Place images of known people inside the images/ folder.
* The filenames of images act as labels for recognition.
* Example: An image named bill\_gates.jpg will be recognized as "Bill Gates".

**2. Encoding Faces (encodings.py)**

* This script:
  + Scans the images/ folder and extracts facial features.
  + Converts faces into numerical encodings using deep learning.
  + Saves these encodings in dataset/face\_encodings.pkl.

**3. Running Face Recognition (face\_recognition.py)**

* The script:
  + Loads stored encodings from face\_encodings.pkl.
  + Starts the webcam and captures live video frames.
  + Converts each frame to grayscale.
  + Uses **Haar Cascades** to detect faces.

**4. Matching Detected Faces**

* If a face is detected:
  + The system crops the face and converts it to RGB.
  + Generates a new encoding for the detected face.
  + Compares it with stored encodings.
  + If a match is found, it displays the person's name.
  + If no match is found, it labels the person as "Unknown".

**5. Displaying Results**

* A **green box** is drawn around detected faces.
* The **recognized name** is displayed in **green text**.
* If the person is unknown, the text still appears, but it says **"Unknown"**.

**6. Stopping the Program**

* The program runs in a loop until the **ESC** key is pressed.
* Once stopped, the webcam feed is released, and all OpenCV windows close.

**How Haar Cascades & Deep Learning Work Together**

🔹 Haar Cascades → Used for fast face detection.

🔹 Deep Learning (face\_recognition library) → Used for face recognition and matching.

✅ Combining both methods ensures a fast and accurate face recognition system.