Facial recognition system for security﻿import cv2﻿import face\_recognition﻿import os﻿import numpy as np﻿# Load known faces﻿known\_face\_encodings = []﻿known\_face\_names = []﻿path = 'known\_faces'  # Folder with known images﻿for filename in os.listdir(path):﻿    img = face\_recognition.load\_image\_file(f"{path}/{filename}")﻿    encoding = face\_recognition.face\_encodings(img)[0]﻿    known\_face\_encodings.append(encoding)﻿    known\_face\_names.append(os.path.splitext(filename)[0])﻿# Initialize webcam﻿video\_capture = cv2.VideoCapture(0)﻿while True:﻿    ret, frame = video\_capture.read()﻿    small\_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)﻿    rgb\_small\_frame = small\_frame[:, :, ::-1]﻿    # Detect faces﻿    face\_locations = face\_recognition.face\_locations(rgb\_small\_frame)﻿    face\_encodings = face\_recognition.face\_encodings(rgb\_small\_frame, face\_locations)﻿    for (top, right, bottom, left), face\_encoding in zip(face\_locations, face\_encodings):﻿        # Compare with known faces﻿        matches = face\_[recognition.com](recognition.com#recognition.com)pare\_faces(known\_face\_encodings, face\_encoding)﻿        name = "Unknown"﻿        face\_distances = face\_recognition.face\_distance(known\_face\_encodings, face\_encoding)﻿        if len(face\_distances) > 0:﻿            best\_match\_index = np.argmin(face\_distances)﻿            if matches[best\_match\_index]:﻿                name = known\_face\_names[best\_match\_index]﻿        # Scale face locations back﻿        top \*= 4﻿        right \*= 4﻿        bottom \*= 4﻿        left \*= 4﻿        # Draw box and label﻿        cv2.rectangle(frame, (left, top), (right, bottom), (0, 255, 0), 2)﻿        cv2.putText(frame, name, (left, top - 10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.75, (0, 255, 0), 2)﻿    cv2.imshow('Security Feed', frame)﻿    if cv2.waitKey(1) & 0xFF == ord('q'):﻿        break﻿video\_capture.release()﻿cv2.destroyAllWindows()