

Face Detection and Recognition System

This AI application detects and recognizes faces in images using OpenCV and the face_recognition library.

It uses pre-trained models to detect face boundaries and extract facial features, then compares them with known faces.

1. Import Libraries

```
import cv2
import face_recognition
import os
```

2. Load Known Faces

```
known_faces_dir = "known_faces"
known_encodings = []
known_names = []

for filename in os.listdir(known_faces_dir):
    image_path = os.path.join(known_faces_dir, filename)
    image = face_recognition.load_image_file(image_path)
    encoding = face_recognition.face_encodings(image)[0]
    known_encodings.append(encoding)
    known_names.append(os.path.splitext(filename)[0])
```

3. Load and Process Test Image

```
test_image = face_recognition.load_image_file("test_image.jpg")
face_locations = face_recognition.face_locations(test_image)
face_encodings = face_recognition.face_encodings(test_image, face_locations)
```

4. Compare Faces and Draw Boxes

```
image_bgr = cv2.cvtColor(test_image, cv2.COLOR_RGB2BGR)

for (top, right, bottom, left), face_encoding in zip(face_locations, face_encodings):
    matches = face_recognition.compare_faces(known_encodings, face_encoding)
    name = "Unknown"

    if True in matches:
        match_index = matches.index(True)
        name = known_names[match_index]
```

```
cv2.rectangle(image_bgr, (left, top), (right, bottom), (0, 255, 0), 2)
cv2.putText(image_bgr, name, (left, top - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255,
255, 255), 2)
```

5. Display or Save Result

```
cv2.imshow("Face Recognition", image_bgr)
cv2.waitKey(0)
cv2.destroyAllWindows()
# To save instead:
# cv2.imwrite("output.jpg", image_bgr)
```

The output will display the test image with recognized faces labeled. If no match is found, it will label the face as 'Unknown'. Ensure the 'known_faces/' folder contains labeled images (e.g., john.jpg).