```
!pip install opencv-python   # checking installedlibarary tools in google colab
!pip install opencv-contrib-python
```

```
Requirement already satisfied: opencv-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.11/dist-packages (from opencv-python) (2.0.2)
Requirement already satisfied: opencv-contrib-python in /usr/local/lib/python3.11/dist-packages (4.11.0.86)
Requirement already satisfied: numpy>=1.21.2 in /usr/local/lib/python3.11/dist-packages (from opencv-contrib-python) (2.0.2)
```

```
from google.colab import files
print("📤 Upload your training image (user1.jpg) and test image (test.jpg)")
uploaded = files.upload()
```

```
📤 Upload your training image (user1.jpg) and test image (test.jpg)
```

Choose Files 2 files

- **test.jpg**(image/jpeg) - 493829 bytes, last modified: 7/22/2025 - 100% done
- **user1.jpg**(image/jpeg) - 482650 bytes, last modified: 7/22/2025 - 100% done

```
Saving test.jpg to test (3).jpg
Saving user1.jpg to user1 (3).jpg
```

```
import os
print("✅ Uploaded Files:", os.listdir())
```

```
✅ Uploaded Files: ['.config', 'test (2).jpg', 'user1 (3).jpg', 'known_user.jpg.jpg', 'user1.jpg', 'user1 (1).jpg', 'user1 (2).jpg', 't
```

```
import cv2
import numpy as np

# Load your training image
train_img = cv2.imread("user1.jpg", cv2.IMREAD_GRAYSCALE)
if train_img is None:
    raise Exception("❌ Error: 'user1.jpg' not loaded. Please re-upload.")

# Resize image
train_img = cv2.resize(train_img, (200, 200))

# Prepare training data
faces = [train_img]
labels = [0]  # md zeyad = label 0

# Create and train the recognizer
recognizer = cv2.face.LBPHFaceRecognizer_create()
recognizer.train(faces, np.array(labels))

print("✅ Face recognizer trained with md zeyad's image.")
```

```
✅ Face recognizer trained with md zeyad's image.
```

```
# Load the test image
test_img = cv2.imread("test.jpg", cv2.IMREAD_GRAYSCALE)
if test_img is None:
    raise Exception("❌ Error: 'test.jpg' not loaded. Please re-upload.")

# Resize test image
test_img = cv2.resize(test_img, (200, 200))

# Predict
label, confidence = recognizer.predict(test_img)

print(f"🔍 Prediction confidence: {confidence:.2f}")

if confidence < 60:
    print("✅ Access Granted: Welcome, md zeyad")
else:
    print("❌ Access Denied: Face not recognized")
```

```
🔍 Prediction confidence: 36
✅ Access Granted: Welcome,
```

What can I help you build?

```
from matplotlib import pyplot as plt
```

```
# Load original (color) versions for display
user_img = cv2.imread("user1.jpg")
test_img_color = cv2.imread("test.jpg")

# Plot side-by-side
fig, ax = plt.subplots(1, 2, figsize=(10, 4))

ax[0].imshow(cv2.cvtColor(user_img, cv2.COLOR_BGR2RGB))
ax[0].set_title("Training: md zeyad")
ax[0].axis("off")

ax[1].imshow(cv2.cvtColor(test_img_color, cv2.COLOR_BGR2RGB))
ax[1].set_title("Test Image")
ax[1].axis("off")

plt.tight_layout()
plt.show()
```