```
!pip install retina-face
```

```
 kB)
/python3.10/dist-packages (from retina-face) (1.26.4)
/python3.10/dist-packages (from retina-face) (5.2.0)
/python3.10/dist-packages (from retina-face) (10.4.0)
:al/lib/python3.10/dist-packages (from retina-face) (4.10.0.84)
/lib/python3.10/dist-packages (from retina-face) (2.17.0)
›/python3.10/dist-packages (from gdown>=3.10.1->retina-face) (4.12.3)
›n3.10/dist-packages (from gdown>=3.10.1->retina-face) (3.16.1)
ib/python3.10/dist-packages (from gdown>=3.10.1->retina-face) (2.32.3)
l0/dist-packages (from gdown>=3.10.1->retina-face) (4.66.5)
›/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (1.4.0)
/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (1.6.3)
:al/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (24.3.25)
.2.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-f
al/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (0.2.0)
›ython3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (3.11.0)
lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (18.1.1)
/local/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (0.4.1)
/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (3.3.0)
›on3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (24.1)
l.2,!=4.21.3,!=4.21.4,!=4.21.5,<5.0.0dev,>=3.20.3 in /usr/local/lib/python3.10/di
:hon3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (71.0.4)
/thon3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (1.16.0)
lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (2.4.0)
·/local/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (4.12.
/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (1.16.0)
al/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (1.64.1)
/local/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (2.17.0
›ython3.10/dist-packages (from tensorflow>=1.9.0->retina-face) (3.4.1)
.23.1 in /usr/local/lib/python3.10/dist-packages (from tensorflow>=1.9.0->retina-
l/lib/python3.10/dist-packages (from astunparse>=1.6.0->tensorflow>=1.9.0->retina
l0/dist-packages (from keras>=3.2.0->tensorflow>=1.9.0->retina-face) (13.8.1)
.10/dist-packages (from keras>=3.2.0->tensorflow>=1.9.0->retina-face) (0.0.8)
3.10/dist-packages (from keras>=3.2.0->tensorflow>=1.9.0->retina-face) (0.12.1)
·/local/lib/python3.10/dist-packages (from requests[socks]->gdown>=3.10.1->retina
›ython3.10/dist-packages (from requests[socks]->gdown>=3.10.1->retina-face) (3.10
l/lib/python3.10/dist-packages (from requests[socks]->gdown>=3.10.1->retina-face)
l/lib/python3.10/dist-packages (from requests[socks]->gdown>=3.10.1->retina-face)
ib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->tensorflow>=1.9.0->ret
=0.7.0 in /usr/local/lib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->
ib/python3.10/dist-packages (from tensorboard<2.18,>=2.17->tensorflow>=1.9.0->ret
/python3.10/dist-packages (from beautifulsoup4->gdown>=3.10.1->retina-face) (2.6)
local/lib/python3.10/dist-packages (from requests[socks]->gdown>=3.10.1->retina-f
/lib/python3.10/dist-packages (from werkzeug>=1.0.1->tensorboard<2.18,>=2.17->ten
›cal/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow>=1.9.0->re
/local/lib/python3.10/dist-packages (from rich->keras>=3.2.0->tensorflow>=1.9.0->
:hon3.10/dist-packages (from markdown-it-py>=2.2.0->rich->keras>=3.2.0->tensorflo
```

```
import cv2
from retinaface import RetinaFace
# Load the image
image_path = '/content/bti.jpg'  # Replace with the path to your image
image = cv2.imread(image_path)
# Detect faces in the image
faces = RetinaFace.detect_faces(image)

# Check if any face is detected
if len(faces) == 0:
    print("No face detected.")
else:
    print(f"Detected {len(faces)} face(s).")

# Extract the face based on
#the detected bounding box
for key, face in faces.items():
    # Get bounding box coordinates
    facial_area = face['facial_area']
    x1, y1, x2, y2 = facial_area

    # Crop the face from the image
    extracted_face = image[y1:y2, x1:x2]

    # Save the extracted face
    output_path = 'extracted_face_dlb.jpg'
    cv2.imwrite(output_path, extracted_face)
    print(f"Face extracted and saved at {output_path}.")
```

xtracted_face.jpg extracted_face.jpg.jpg ×

```
    # Optional: Display the extracted face
    # cv2.imshow("Extracted Face", extracted_face)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

```
No face detected.
```

```
import cv2
import numpy as np
from insightface.app import FaceAnalysis

# Step 1: Initialize and prepare the ArcFace model
app = FaceAnalysis(providers=['CUDAExecutionProvider', 'CPUExecutionProvider'])  # Us
app.prepare(ctx_id=0, det_size=(640, 640))  # ctx_id=0 uses GPU; -1 uses CPU

def preprocess_image(image):
    """
    Preprocess the image by resizing and normalizing it to match the model's input re
    """
    # Resize image to the model's expected input size (112x112 for ArcFace)
    resized_image = cv2.resize(image, (112, 112))

    # Normalize the image: scale pixel values to the range [0, 1]
    normalized_image = resized_image.astype(np.float32) / 255.0

    # Convert the image to RGB format (OpenCV loads images in BGR format)
    rgb_image = cv2.cvtColor(normalized_image, cv2.COLOR_BGR2RGB)

    return rgb_image

def extract_features(image_path):
    """
    Extracts face embeddings from an image using the ArcFace model.
    """
    # Load the image
    image = cv2.imread(image_path)
    if image is None:
        print("Error: Image not loaded. Please check the path.")
        return None

    # Preprocess the image (resize and normalize)

    preprocessed_image = preprocess_image(image)
    output_path = f'{image_path}.jpg'
    cv2.imwrite(output_path, preprocessed_image)

    # Detect faces in the preprocessed image
    faces = app.get(preprocessed_image)
    if len(faces) != 1:
        print(f"Expected 1 face, but found {len(faces)} face(s).")
        return None

    # Extract the face embedding (feature vector)
    embedding = faces[0].embedding
    return embedding

def compare_faces(embedding1, embedding2):
    """
    Compares two face embeddings using cosine similarity.
    """
    # Compute cosine similarity
    similarity = np.dot(embedding1, embedding2) / (np.linalg.norm(embedding1) * np.li
    return similarity

# Provide paths to the two face images
image_path_1 = '/content/Rohan-pic.jpg'  # Path to the first face image
image_path_2 = '/content/Rohan-pic.jpg'         # Path to the second face image

# Step 2: Extract features from both images
embedding1 = extract_features(image_path_1)
embedding2 = extract_features(image_path_2)

# Step 3: Compare the embeddings if extraction was successful
if embedding1 is not None and embedding2 is not None:
    similarity_score = compare_faces(embedding1, embedding2)
    print(f"Similarity Score: {similarity_score}")

    # Define a threshold to determine if the faces match (usually around 0.5 - 0.6)
    threshold = 0.6
    if similarity_score > threshold:
```

```
        print(f"The faces match with a similarity score of {similarity_score:.4f}.")
    else:
        print(f"The faces do not match with a similarity score of {similarity_score:.
else:
    print("Feature extraction failed for one or both images.")
```

```
Applied providers: ['CPUExecutionProvider'], with options: {'CPUExecutionProvide
find model: /root/.insightface/models/buffalo_l/1k3d68.onnx landmark_3d_68 ['Nor
Applied providers: ['CPUExecutionProvider'], with options: {'CPUExecutionProvide
find model: /root/.insightface/models/buffalo_l/2d106det.onnx landmark_2d_106 [
Applied providers: ['CPUExecutionProvider'], with options: {'CPUExecutionProvide
find model: /root/.insightface/models/buffalo_l/det_10g.onnx detection [1, 3, '
Applied providers: ['CPUExecutionProvider'], with options: {'CPUExecutionProvide
find model: /root/.insightface/models/buffalo_l/genderage.onnx genderage ['None
Applied providers: ['CPUExecutionProvider'], with options: {'CPUExecutionProvide
find model: /root/.insightface/models/buffalo_l/w600k_r50.onnx recognition ['Nor
set det-size: (640, 640)
Expected 1 face, but found 0 face(s).
Expected 1 face, but found 0 face(s).
Feature extraction failed for one or both images.
```

```
!pip install insightface
```

```
Collecting insightface
  Downloading insightface-0.7.3.tar.gz (439 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 439.5/439.5 kB 7.5 MB/s eta 0:0
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packag
Collecting onnx (from insightface)
  Downloading onnx-1.16.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-package
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-p
Requirement already satisfied: Pillow in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist
Requirement already satisfied: scikit-image in /usr/local/lib/python3.10/dist
Requirement already satisfied: easydict in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: cython in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: albumentations in /usr/local/lib/python3.10/di
Requirement already satisfied: prettytable in /usr/local/lib/python3.10/dist-
Requirement already satisfied: PyYAML in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: pydantic>=2.7.0 in /usr/local/lib/python3.10/d
Requirement already satisfied: albucore>=0.0.15 in /usr/local/lib/python3.10/
Requirement already satisfied: eval-type-backport in /usr/local/lib/python3.1
Requirement already satisfied: opencv-python-headless>=4.9.0.80 in /usr/local
Requirement already satisfied: networkx>=2.8 in /usr/local/lib/python3.10/dis
Requirement already satisfied: imageio>=2.33 in /usr/local/lib/python3.10/dis
Requirement already satisfied: tifffile>=2022.8.12 in /usr/local/lib/python3.
Requirement already satisfied: packaging>=21 in /usr/local/lib/python3.10/dis
Requirement already satisfied: lazy-loader>=0.4 in /usr/local/lib/python3.10/
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3
Requirement already satisfied: protobuf>=3.20.2 in /usr/local/lib/python3.10/
Requirement already satisfied: wcwidth in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/pyt
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.1
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.1
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/pythc
Requirement already satisfied: pydantic-core==2.23.4 in /usr/local/lib/pythor
Requirement already satisfied: typing-extensions>=4.6.1 in /usr/local/lib/pyt
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-pac
Downloading onnx-1.16.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_6
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 15.9/15.9 MB 77.4 MB/s eta 0:00:0
Building wheels for collected packages: insightface
  Building wheel for insightface (pyproject.toml) ... done
  Created wheel for insightface: filename=insightface-0.7.3-cp310-cp310-linux
  Stored in directory: /root/.cache/pip/wheels/e3/d0/80/e3773fb8b6d1cca87ea1c
Successfully built insightface
Installing collected packages: onnx, insightface
Successfully installed insightface-0.7.3 onnx-1.16.2
```

```
!pip install onnxruntime
```

```
Collecting onnxruntime
  Downloading onnxruntime-1.19.2-cp310-cp310-manylinux_2_27_x86_64.manylinux_2_
Collecting coloredlogs (from onnxruntime)
  Downloading coloredlogs-15.0.1-py2.py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: flatbuffers in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: numpy>=1.21.6 in /usr/local/lib/python3.10/dist-
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-packa
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages
Collecting humanfriendly>=9.1 (from coloredlogs->onnxruntime)
  Downloading humanfriendly-10.0-py2.py3-none-any.whl.metadata (9.2 kB)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/
Downloading onnxruntime-1.19.2-cp310-cp310-manylinux_2_27_x86_64.manylinux_2_28_
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.2/13.2 MB 77.1 MB/s eta 0:00:00
Downloading coloredlogs-15.0.1-py2.py3-none-any.whl (46 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 46.0/46.0 kB 2.8 MB/s eta 0:00:00
Downloading humanfriendly-10.0-py2.py3-none-any.whl (86 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 86.8/86.8 kB 5.9 MB/s eta 0:00:00
Installing collected packages: humanfriendly, coloredlogs, onnxruntime
Successfully installed coloredlogs-15.0.1 humanfriendly-10.0 onnxruntime-1.19.2
```

```
import cv2
import numpy as np
from keras.models import load_model
from keras.preprocessing import image

# Load the pre-trained FaceNet model
model = load_model('/content/facenet_keras.h5')

def preprocess_image(img_path):
    # Load the image
    img = cv2.imread(img_path)
    img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)  # Convert BGR to RGB
    img = cv2.resize(img, (160, 160))  # Resize to 160x160
    img = img.astype('float32')
    img = img / 255.0  # Normalize the image
    img = np.expand_dims(img, axis=0)  # Add batch dimension
    return img

def get_embedding(img_path):
    # Preprocess the image and get the embedding
    preprocessed_image = preprocess_image(img_path)
    embedding = model.predict(preprocessed_image)
    return embedding

def compare_faces(embedding1, embedding2, threshold=0.5):
    # Calculate the Euclidean distance between the embeddings
    distance = np.linalg.norm(embedding1 - embedding2)
    print(f'Distance: {distance}')

    if distance < threshold:
        print("The faces match.")
    else:
        print("The faces do not match.")

# Provide paths to the two face images
image_path_1 = 'path_to_image_1.jpg'  # Path to the first face image
image_path_2 = 'path_to_image_2.jpg'  # Path to the second face image

# Get embeddings for both images
embedding1 = get_embedding(image_path_1)
embedding2 = get_embedding(image_path_2)

# Compare the two embeddings
compare_faces(embedding1, embedding2)
```

```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-27-3b95505f3d0f> in <cell line: 7>()
      5
      6 # Load the pre-trained FaceNet model
----> 7 model = load_model('/content/facenet_keras.h5')
      8
      9 def preprocess_image(img_path):

                                 10 frames
/usr/local/lib/python3.10/dist-packages/keras/src/utils/python_utils.py in
func_load(code, defaults, closure, globs)

ValueError: bad marshal data (unknown type code)
```

Next steps: Explain error

```
import os
print(os.path.exists('/content/facenet_keras.h5'))
```

```
True
```

```
!pip install keras==2.4.3
!pip install tensorflow==2.4.1
```

```
Requirement already satisfied: keras==2.4.3 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: numpy>=1.9.1 in /usr/local/lib/python3.10/dist-pa
Requirement already satisfied: scipy>=0.14 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: pyyaml in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: h5py in /usr/local/lib/python3.10/dist-packages
ERROR: Could not find a version that satisfies the requirement tensorflow==2.4.1
ERROR: No matching distribution found for tensorflow==2.4.1
```

Start coding or generate with AI.

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.