Facial Recognition

By Salvador Leon Lopez

Requirements:

cmake
dlib
face_recognition
numpy
opencv-python

The Samples

How it works

- Images with “Known faces” are recognized encoded and listed in a directory
- An unknown image is encoded with the faces detected
- Distance between the enconded unknown faces and the recognized faces are compared
- Pick the known face with the least distance to the unknown and display the name

KNOWN FACES CODE

```python
#Get encoded faces to recognize
def get_encoded_faces():
    encoded = {}
    for dirpath, dnames, fnames in os.walk("./faces"):
        for picture in fnames:
            if picture.endswith(".jpg") or picture.endswith(".png") or picture.endswith(".jpeg"):
                face = fr.load_image_file("faces/" + picture)
                print(picture)
                encoding = fr.face_encodings(face)[0]
                encoded[picture.split(".")[0]] = encoding
    return encoded
```

```python
def classify_face(im):
#Check if a face in the image is already known and clasify them
    faces = get_encoded_faces()
    faces_encoded = list(faces.values())
    known_face_names = list(faces.keys())

    img = cv2.imread(im, 1)

    face_locations = face_recognition.face_locations(img)
    unknown_face_encodings = face_recognition.face_encodings(img, face_locations)

    face_names = []
    for face_encoding in unknown_face_encodings:
        matches = face_recognition.compare_faces(faces_encoded, face_encoding)
        name = "Unknown"

        face_distances = face_recognition.face_distance(faces_encoded, face_encoding)
        best_match_index = np.argmin(face_distances)
        if matches[best_match_index]:
            name = known_face_names[best_match_index]

        face_names.append(name)
```

UNKNOWN FACES CODE

Results