**Facial recognition and OCR**

1. **Facial recognition**

Validation of identical people by comparing two photos (existing APIs – existing models – libraries used for existing solutions)

* [Face API](https://azure.microsoft.com/en-us/services/cognitive-services/face/) - Microsoft Azure developer tool, free 12 month trial with an Azure account. Verify identity or control access with ready-for-scale face recognition.
  + 0-1M transactions - **$1** per 1,000 transactions
  + 1-5M transactions - **$0.80** per 1,000 transactions
* [Amazon Rekognition](https://aws.amazon.com/getting-started/hands-on/detect-analyze-compare-faces-rekognition/) - developer API with free 12-month trial as part of the AWS Free Tier. Vous utiliserez [Amazon Rekognition](https://aws.amazon.com/rekognition/) pour analyser et comparer des images afin de définir si les visages qui s’y trouvent sont les mêmes.  Au cours de la période d'offre gratuite, vous pouvez analyser jusqu'à 5 000 images par mois gratuitement.
  + Premier million d'images : 0,001 USD par image
  + 4 millions d’images suivants : 0,0008 USD par image
* MXFACE

MXFACE Is a leading technology brand that brings cutting-edge facial recognition APIs for a seamless application experience. <https://mxface.ai/face-API-Price>

Its free trial contains:

* 100 API per day
* [**Face Detect API**](https://mxface.ai/facedetection)
* [**Face Landmark API**](https://mxface.ai/facelandmarks)
* [**Face Compare API**](https://mxface.ai/facecomparing)
* [**Face Attributes API**](https://mxface.ai/faceattributes)
* Veriff : $49 monthly min commitment

<https://www.veriff.com/product/identity-verification>

* Bio Id playground : for free

<https://playground.bioid.com/PhotoVerify>

* Face similarity test online \*\*
* <https://facecomparison.toolpie.com/>
* <https://www.cogniware.com/en/insights/facial-recognition-1>
* <https://www.faceplusplus.com/face-comparing/>

We’ll take a CNN that had been previously trained for face recognition on a dataset with millions of images – and adapt it to solve our problem.

**FACEPP**

the module **python-facepp** can be used for doing the Face Comparison. This module is for communicating with Face++ facial recognition service.

This app compares two photographs of the same person or two different persons against his/her face features like face landmarks, beauty score, face emotion, etc. If both photographs are matching with each other, the app result is **“Both photographs are of same person”**otherwise app result is **“Both photographs are of two different persons”.**

pip install python-facepp

# Face-comparison

AI Face comparison using FaceNet, compare two photos and see if they are the same person.

pip install face-compare

The [opencv library](https://pypi.org/project/opencv-python/" \t "_blank), is an open source computer vision and machine learning software library.

import imutils  
import face\_recognition

**FACE DEEP:**

A Lightweight Face Recognition and Facial Attribute Analysis Framework (Age, Gender, Emotion, Race) for Python

Deepface also comes with a strong facial attribute analysis module including [age](https://sefiks.com/2019/02/13/apparent-age-and-gender-prediction-in-keras/), [gender](https://sefiks.com/2019/02/13/apparent-age-and-gender-prediction-in-keras/), [facial expression](https://sefiks.com/2018/01/01/facial-expression-recognition-with-keras/) (including angry, fear, neutral, sad, disgust, happy and surprise) and [race](https://sefiks.com/2019/11/11/race-and-ethnicity-prediction-in-keras/) (including asian, white, middle eastern, indian, latino and black) predictions.

**FacePP :**

Python-FacePP is a library for communicating with a [Face++](https://www.faceplusplus.com/) facial recognition service. Face++ exposes some of its data via [Web API](https://console.faceplusplus.com/documents/6329584) for which Python-FacePP provides a simple but powerful Pythonic API

1. **OCR**

Optical Character Recognition is a widespread technology to recognize text inside images, such as scanned documents and photos. OCR technology is used to convert virtually any kind of image containing written text (typed, handwritten, or printed) into machine-readable text data.

OCR results depend on the input data quality. A clean segmentation of the text and no noise in the background gives better results. In the real world, this is not always possible, so we need to apply multiple pre-processing techniques for OCR to give better results.

* OCR gratuit avec Keep Google : <https://keep.google.com/>
* Fairy text
* OCR GRATUIT :
  + <https://www.imagetotext.info/>
  + <https://www.ocrnow.com/result>
  + <https://www.ocr.best/image-to-text>

Les outils pour faire le OCR :

* Keras OCR
* Easy OCR
* Pytesseract

### **\* Keras-OCR** is image specific OCR tool. If text is inside the image and their fonts and colors are unorganized, Keras-ocr consumes time if used on CPU.

### \* **EasyOCR** is lightweight model which is giving a good performance for receipt or PDF conversion. It is giving more accurate results with organized texts like pdf files, receipts, bills. EasyOCR also performs well on noisy images.

### \* **Pytesseract**

### Python-tesseract is an optical character recognition (OCR) tool for python. That is, it will recognize and “read” the text embedded in images.

Tesseract is an open-source OCR Engine, managed by Google.

### is performing well for high-resolution images. Certain morphological operations such as dilation, erosion, OTSU binarization can help increase pytesseract performance. It also provides better results on handwritten text as compared to EasyOCR.