



Biometric Technologies and Behavioural Security

Tutorial 4 - Face Analysis and Recognition

Face recognition

NON-INVASIVE

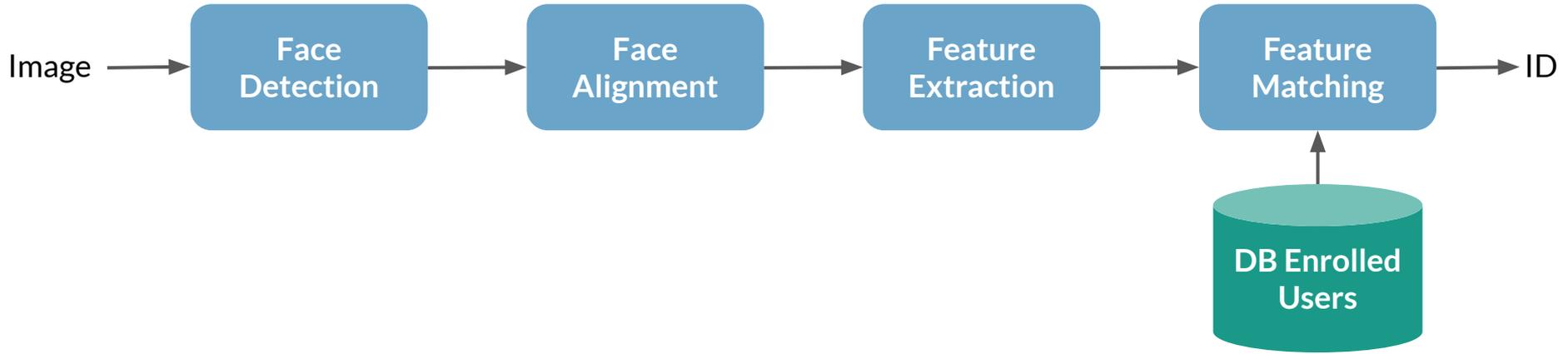
A HIGH LEVEL OF UNIQUENESS

INTUITIVE





Face recognition processing flow





Face Detection Techniques

Feature-based

Appearance-based

Knowledge-based

Template matching

- **Haar Cascades Classifier:** The first machine learning-based cascading classifier to fulfill the requirement of fast implementation on low-power CPUs, such as cameras and phones.
- **Histogram of Oriented Gradients (HOG):** Histogram of Oriented Gradient is used as features followed by linear Support Vector Machine classifier.
- **Multi-task Cascaded Convolutional Networks (MTCNN):** Deep learning-based face detection technique.



Haar Cascades Classifier: the Viola-Jones object detection



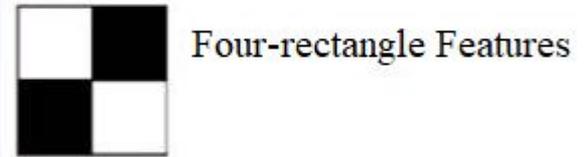
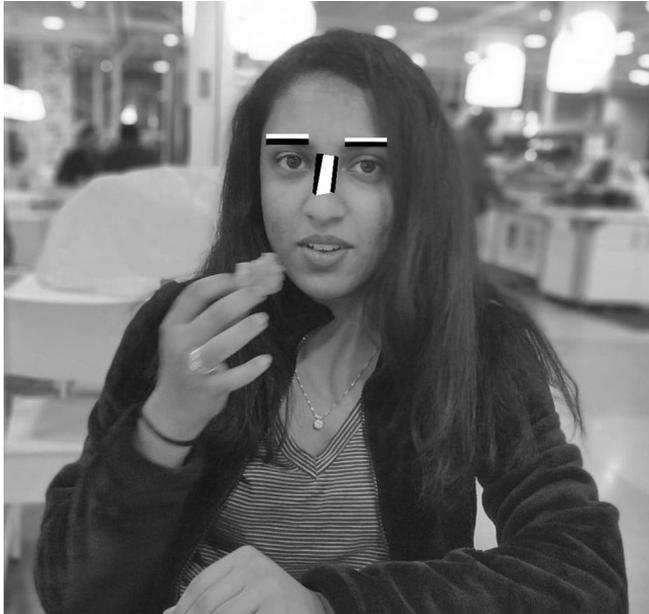
Haar-like
features

AdaBoost
training

Cascading
classifier

Viola-Jones object detection

Haar Features:



Important Features for Face Detection

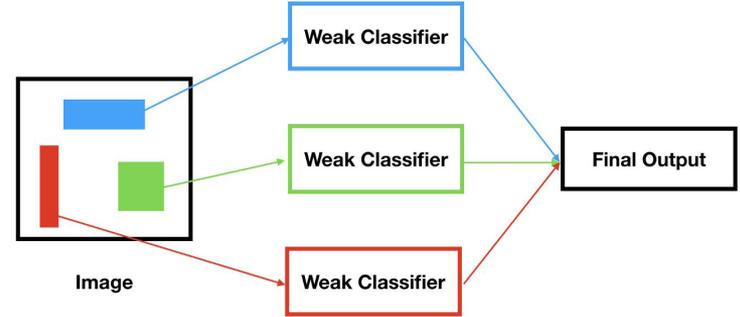


Viola-Jones object detection

Cascading classifier

A series of classifiers are applied to every sub-window.

- increase accuracy
- lower the percentage of false alarms





Viola-Jones object detection (OpenCV)

More info:

<https://towardsdatascience.com/the-intuition-behind-facial-detection-the-viola-jones-algorithm-29d9106b6999>

In OpenCV, this whole model is already pre-trained for face detection:

<https://bit.ly/2VEP27d>



Histogram of Oriented Gradients (HOG) in Dlib

The features extracted are the distribution (histograms) of directions of gradients (oriented gradients) of the image. Gradients are typically large around edges and corners and allow us to detect those regions.



The gradient of an image typically removes non-essential information.



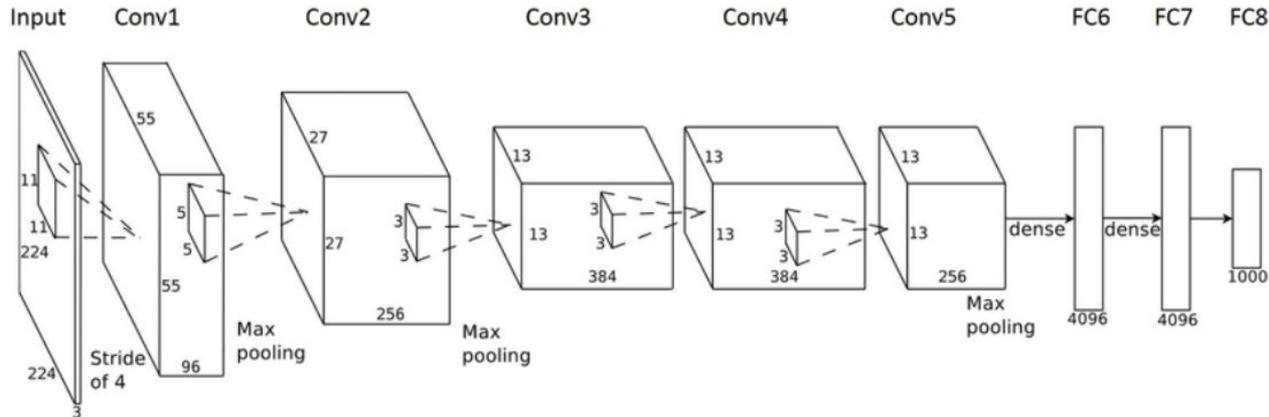
Histogram of Oriented Gradients (HOG) in Dlib

<https://bit.ly/3bFNxLG>



Convolutional Neural Network in Dlib

In previous approaches, a great part of the work was to select the filters in order to create the features in order to extract as much information from the image as possible. With the rise of deep learning and greater computation capacities, this work can now be automated.





Convolutional Neural Network in Dlib

The first step is to download the pre-trained model here:

https://github.com/davisking/dlib-models/blob/master/mmod_human_face_detector.dat.bz2

```
dnnFaceDetector = dlib.cnn_face_detection_model_v1("mmod_human_face_detector.dat")
rects = dnnFaceDetector(gray, 1)
for (i, rect) in enumerate(rects):
    x1 = rect.rect.left()
    y1 = rect.rect.top()
    x2 = rect.rect.right()
    y2 = rect.rect.bottom() # Rectangle around the face
    cv2.rectangle(gray, (x1, y1), (x2, y2), (255, 255, 255), 3)plt.figure(figsize=(12,8))
plt.imshow(gray, cmap='gray')
plt.show()
```



Attention! The `cnn_face_detection_model_v1` method needs CUDA

RuntimeError: Error while calling `cudaGetDevice(&the_device_id)` in file `/tmp/pip-wheel-b9ck8go0/dlib/dlib/cuda/gpu_data.cpp:201`. code: 100, reason: no CUDA-capable device is detected

CUDA (Compute Unified Device Architecture) is a parallel computing platform and application programming interface (API) model created by Nvidia.

CUDA code doesn't run on AMD CPU or Intel HD graphics unless you have a NVIDIA Hardware inside you Machine

[Click to Runtime > Change runtime type> Hardware Accelerator GPU.](#)



Convolutional Neural Network in Dlib

<https://bit.ly/3f2N8F7>



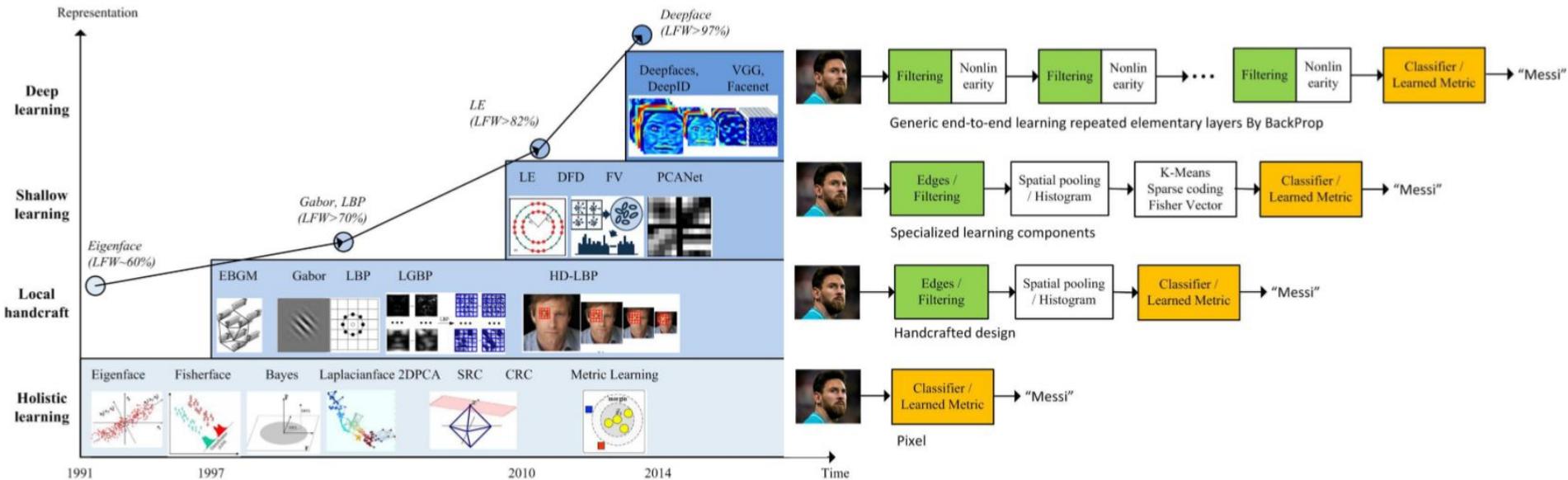
Face Alignment/normalization

Faces should be:

- centered in the image;
- rotated such that the eyes lie on a horizontal line (i.e., the face is rotated such that the eyes lie along the same y-coordinates);
- scaled such that the size of the faces are approximately identical.

<https://bit.ly/2Y6yBSN>

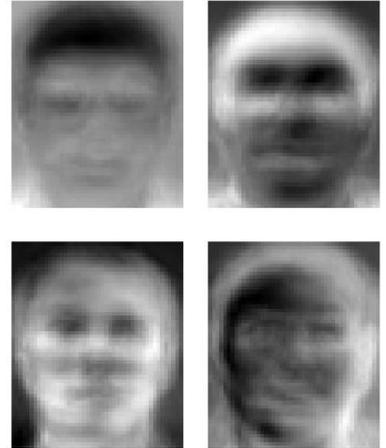
Milestones of feature representation for FR



EigenFace Recognition

- A mathematical procedure performs a dimensionality reduction by extracting the principal component of multi-dimensional data.
- The eigenvectors are derived from the covariance matrix of the probability distribution over the high-dimensional vector space of face images.

<https://bit.ly/3bMNgXm>





Try LBP and BSIF for Face Recognition

https://colab.research.google.com/drive/1ASVVL1aTPxrnq_oEI_QiDX7e0hljMk7w