DLCP Sep'18 Project-2

The Real Problem

What is Face recognition?

"Say hello to the future" the tagline of iPhoneX marked the advent of face recognition into mainstream apps using it as a feature to unlock the phone. This marks a milestone in itself as far as facial recognition technology is concerned. - Deepa Naik

Facial recognition is a category of biometric software that maps an individual's facial features mathematically and stores the data as a faceprint. The software uses deep learning algorithms to compare a live capture or digital image to the stored faceprint in order to verify an individual's identity.

Facial recognition is being used in many businesses

You're used to unlocking your door with a key, but maybe not with your face. As strange as it sounds, our physical appearances can now verify payments, grant access and improve existing security systems. Protecting physical and digital possessions is a universal concern which benefits everyone unless you're a cybercriminal or a kleptomaniac of course. Facial biometrics are gradually being applied to more industries, disrupting design, manufacturing, construction, law enforcement, and healthcare.

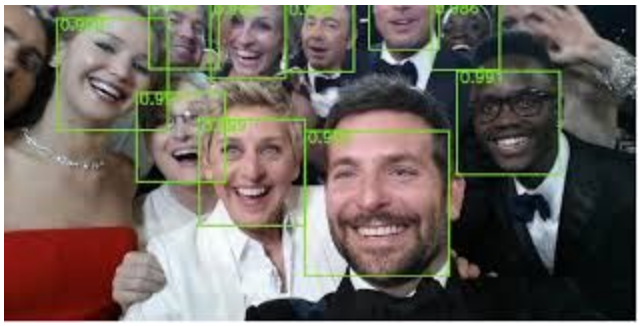
Understanding Face Recognition Software

Face recognition deals with Computer Vision a discipline of Artificial Intelligence and uses techniques of image processing and deep learning. Face recognition algorithms can be further classified based on whether they are used on 2D or 3D images or on finding faces in motion, like in a video.

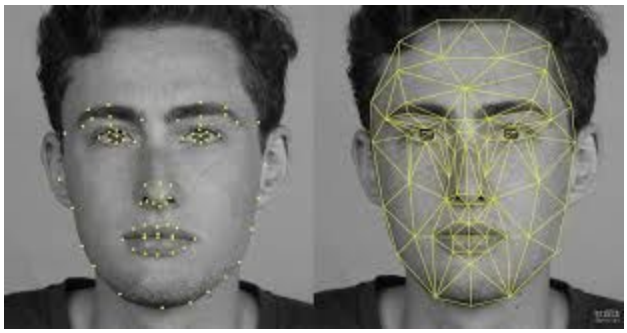
Face Detection vs. Face Recognition

Though they sound similar, the complexity involved in both is vastly different. In Face Detection, the computer recognizes the face within an image and locates its position. If you have used face changer app on Snapchat, you are using face detection. Face recognition deals with identification to establish whose face it is by matching it to an existing face database. You can refer to the images below for the distinction.

Face detection



Face recognition



Apart from identification, other typical features are

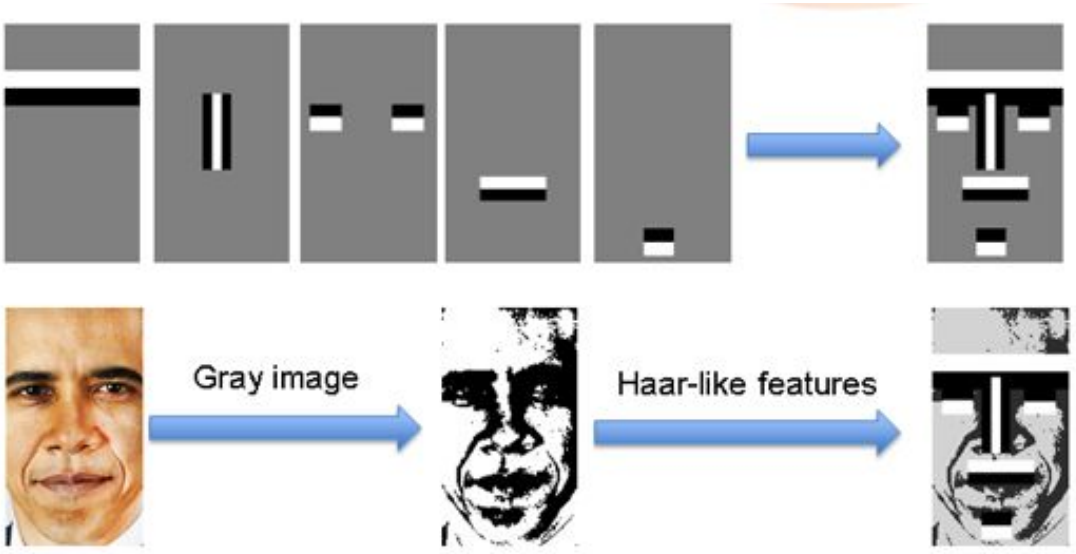
- **Emotion Detection**
- **Age Detection**
- **Gender Detection**
- **Attention Measurement**
- **Sentiment Detection**
- **Ethnicity Detection**

How Does Face Recognition Work?

Until the year 2000, there were many different techniques to detect the face, but all were either slow or unreliable, or both. A major change happened in 2001, when Viola and Jones invented

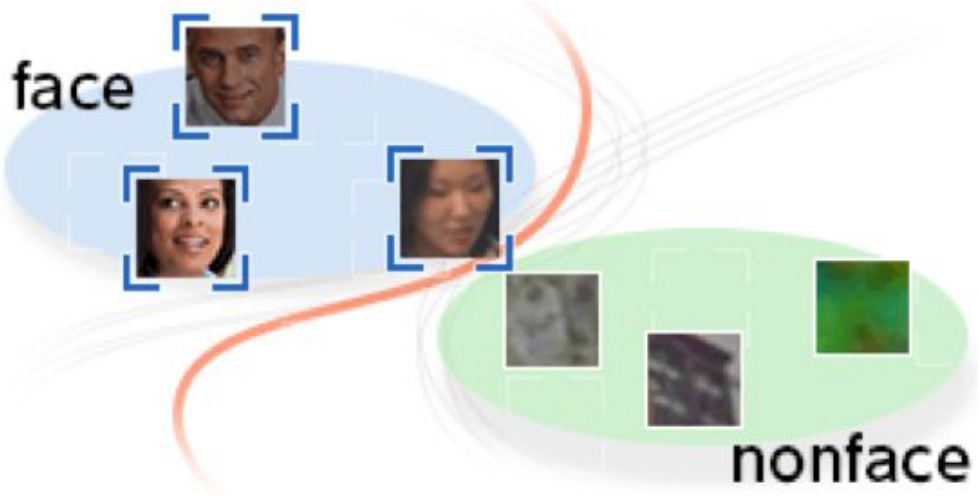
Haar-based cascade classifier, a technique used to identify object and it was improved by Lienhart and Maydt in 2002. The result of identifying objects was fast enough (identifying in real-time on normal PC) and was reliable (more than 95% accuracy).

There are two approaches to the facial recognition: Feature-Based Face Recognition and Appearance-Based Face Recognition. Haar-like features are the rectangle which is divided into different rectangles. First, the image is grayscaled, then the haar-like features (rectangle) are shifted through the image, comparing similar image rectangles with Haar-like features, similar ones are marked.



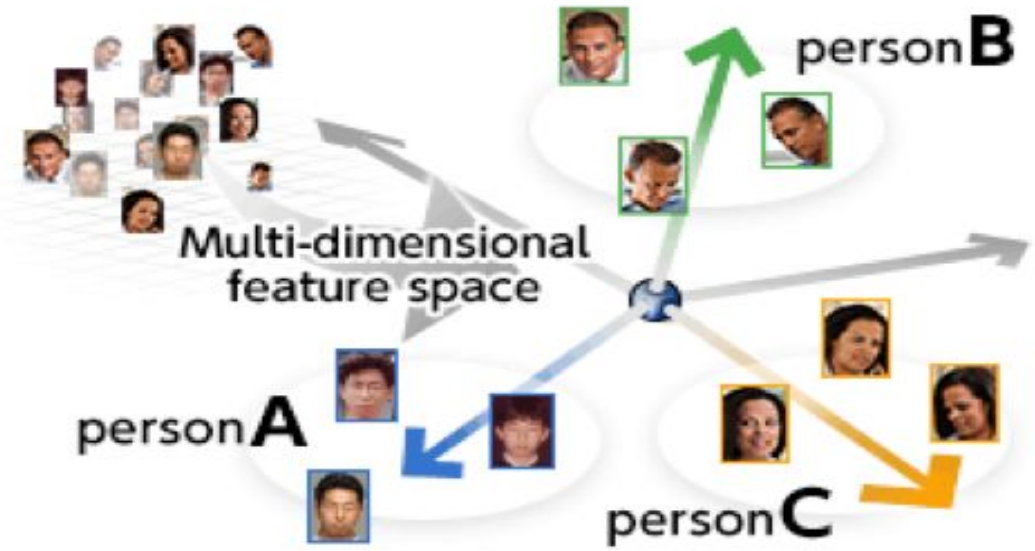
The following are three main steps in the face recognition using haar-like method:

Step 1. Detects the “position of the face”



Step 2. Finds the “feature of the face”

Step 3. Search and identify of the “detected face” in the database.



Project Description

In this hands-on project, the goal is to build a face recognition system, which includes building a face detector to locate the position of a face in an image and a face identification model to recognize whose face it is by matching it to the existing database of faces.

Milestone 2: Face Detector (Note: Milestone 1 is the CNN model on SVHN Dataset that you already executed in week 5)

We will be using transfer learning on an already trained model to build our detector. We will perform transfer learning on MobileNet model which is already trained to perform object detection. We will need to train the last 6-7 layers and freeze the remaining layers to train the model for face detection. To be able to train the MobileNet model for face detection, we will be using WIDER FACE dataset which already has the bounding box data for various images with single face and multiple faces. The output of the model is the bounding box data which gives the location of the face in an image. We learn to build a face detection model using Keras supported by Tensorflow

Milestone 3: Face Identification

In this milestone, we will build a face identification model by using an existing trained model. The CNN model is taken from the Keras-OpenFace project. The architecture details aren't too important here, it's only useful to know that there is a fully connected layer with 128 hidden units followed by an L2 normalization layer on top of the convolutional base. These two top layers are referred to as the embedding layer from which the 128-dimensional embedding vectors can be obtained. Model training aims to learn an embedding f(x) of image x such that the squared L2 distance between all faces of the same identity is small and the distance between a pair of faces from different identities is large. This can be achieved with a triplet loss L that is minimized when the distance between an anchor image and a positive image (same identity) in embedding space is smaller than the distance between that anchor image and a negative image (different identity) by at least a margin.

To demonstrate face recognition on a custom dataset, a small dataset is used. It consists of around 15-25 face images of 10 different persons. The metadata for each image (file and identity name) are loaded into memory for later processing.

Milestone 4: Test the Face Recognition system (the two models together) on a different dataset.

Overview

Milestone 2 overview

In this problem we use "Transfer Learning" of an Object Detector model to detect any object according to the problem in hand.

Here, We are particularly interested in detecting faces in a given image. Below are the steps involved in the project.

1. To use already trained model, we will import the model and its supporting files for the model to function.

2. Set the parameters for the model

3. Import the dataset for the model to train on. For this, we are using the WIDER FACE dataset.

4. Get the labels for these images, so that we can use this information while training for detecting faces with the given model using transfer learning.

5. Making the model ready for training by freezing some layers in which we are not updating the weights while training. We need to train the top 6-7 layers.

6. load the weights of the MobileNet model given in file '''`mobilenet_1_0_224_tf.h5`'''

7. Set the optimizers, loss functions, epochs, learning rate, batch size, checkpointing, early stopping etc..

8. Train the model

9. Load the best saved weights

10. Predict using test images

Instructions for all the above steps are given in the notebook.

Mobilenet explained

https://medium.com/@yu4u/why-mobilenet-and-its-variants-e-g-shufflenet-are-fast-1c7048b9618d

https://hackernoon.com/creating-insanely-fast-image-classifiers-with-mobilenet-in-tensorflow-f030ce0a2991.

MobileNet paper : https://arxiv.org/pdf/1704.04861.pdf

WIDER FACE: A Face Detection Benchmark Dataset

WIDER FACE dataset is a face detection benchmark dataset, of which images are selected from the publicly available WIDER dataset. We choose **32,203** images and label **393,703** faces with a high degree of variability in scale, pose and occlusion as depicted in the sample images. WIDER FACE dataset is organized based on 61 event classes. For each event class, we randomly select 40%/10%/50% data as training, validation and testing sets. We adopt the same evaluation metric employed in the PASCAL VOC dataset. Similar to MALF and Caltech datasets, we do not release bounding box ground truth for the test images. Users are required to submit final prediction files, which we shall proceed to evaluate.

Milestone 3 overview

In this problem we use a pre-trained model trained on Face recognition to recognize similar faces.

Here, We are particularly interested in recognizing whether two given faces are of same person or not. Below are the steps involved in the project.

1. Face Alignment using Dlib
2. Load the dataset and meta information
3. Load the pre-trained model and weights.
4. Generate Embedding vectors for each face in the dataset.
5. Build a distance metrics for identifying the distance between two given images.
6. Build SVM classifier to map each image to its right person.
7. Predict using the SVM model.

Instructions for all the above steps are given in the notebook.

Learn more about Open Face

https://blog.algorithmia.com/understanding-facial-recognition-openface/

Open Face paper :
http://www.cv-foundation.org/openaccess/content_cvpr_2015/app/1A_089.pdf

Project Objectives

The objective of the project is to learn how to do transfer learning and how to use a pre-trained model already trained on the desired task. In this project you are going to use transfer learning to build face detection model and face recognition using pre-trained model The goals of this assignment are as follows:

Milestone 2 : Face Detection

- Learn to how to do transfer learning on a pre-trained model.
- Use transfer learning to build a Face Detection model.
- Learn how to set parameters for the pre-trained model for transfer learning.
- Learn to Set the optimizers, loss functions, epochs, learning rate, batch size, checkpointing, early stopping etc.

Milestone 2 deadline : **Nov 26th, Monday 11:59 pm**
Points : 30

Milestone 3 : Face Identification

- Understand how to use pre-trained models
- Using a pre-trained model to do Face Alignment
- Using a pre-trained model to do Face Recognition.
- Learn to use Triplet Loss.
- Learn to use SVM classifier for prediction.
- Learn to combine both the models to test.

Milestone 3 deadline : **Dec 3rd, Monday 11:59 pm**
Points : 40

Downloads

Refer to Olympus Project 2 module in Week 7 for project related files and instructions.

Working with Datasets in google colabs/Cloud Lab(GPU lab)

- Create a folder called "**DLCP**" in My drive/Cloud.
- Create a "**Project-2**" folder within your **DLCP** folder.
- Store all the datafiles related to the project 2 in the above folder. You can download the datasets from Olympus, week 7 Computer Vision Module.
- Save your data in that folder to access it while working with google colab.
- Download the jupyter notebook " **FACE DETECTION_Questions.ipynb"** from olympus in project 2 section in week 7 to write code for your project milestones. And similarly **"FACE RECOGNITION_Questions.ipynb"** file from week 8 section in Computer Vision module.
- You can rename the file after you have downloaded the notebook as **"Yourname-DLCP-Project2-Milestone2**.ipynb" for project 2 milestone 2 submissions and similarly for all the other submissions. Modify the project number and the milestone number accordingly.
- The jupyter notebook comes with some blocks pre-coded. So please do not change the code in those blocks. Only the blocks that you are needed to code will appear blank and points for the task also will be mentioned next to the title of the block.

Datasets:

Milestone 2:

1. "files_required_for_face_detection.zip"
2. WIDER_train.zip
3. WIDER_val.zip

Milestone 3

4. "files_required_for_face_recognition.zip"

Reference

Acknowledgement for the datasets.
http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/

Project submissions and Evaluation Criteria

While we encourage peer collaboration and contribution, plagiarism, copying the code from other sources or peers will defeat the purpose of coming to this program. We expect highest order of ethical behavior.

Submit the code and the video walkthrough on Olympus.

Submit the project milestone in a jupyter notebook and submit it on olympus for evaluation. The students also need to submit a video walking us through the code and explaining various blocks. Check the project section.

Evaluation Criteria: You need to complete all the tasks mentioned in the notebooks. You must receive a minimum of 60% on each milestone to be complete the project.

>60 % Points = Complete
>80 % Points = Excellent

Project Support

You can clarify your queries related to project during the weekly support sessions conducted by deep learning practitioners on every saturday.

Q&A forum for offline support : piazza
You can also post your queries on the discussion forums piazza for offline support during the rest of the time.
Link to DLCP sep 22 2018 batch : https://piazza.com/class/jma48sr0pee3be?cid=4#

Click on the link and enroll yourself as a student for the class

Alternately you can write to dlcp-support@greatlearning.in. However, we encourage you to post all your queries on piazza for speedy response.

Happy Learning!