Machine Learning Engineer Nanodegree

Capstone Proposal

Luiz Costa

October 31st, 2017

Proposal

Domain Background

Face Recognition is the task of identify or verify a person in a digital image or video stream through his face.

It's an important task for many areas, since social networks, where it can be used to label users in photos, until security, where it can be part of a program that reads a set of photos and try to identify a criminal.

The process can be divided in two phases: Detection and Recognition. For face detection most commonly used algorithm was proposed by Viola & Jones [1]. For face recognition, common algorithms include principal component analysis and eigenfaces.

With the explosion of cell phones with camera, thousands of photos are taken every day. Understand the objects and people in these photos, can play an important role in computer vision applications. In a social media application, a user, when upload a photo to his profile, an algorithm can automatically detect, recognize a person and asking for a user to confirm the label. With a lot of photos tagged, a search engine can take advantage this to find photos where specific persons appear.

My personal motivation for this project, is create a model to recognize famous faces in images. It can easily be extended to a video stream, where it can be used to label actors in films or tv series automatically. One example of this type of application is the ability of tag actors in video of Pornohub, a famous website for adults.[2].

Problem Statement

Who is in the photo? This question summarizes the problem I'm proposing to be solved. Identify a person in image, can help to solve commons problems that we have today. For example, in a airport, is possible to use face recognition to identify criminals wanted by justice. In a search engine, is possible to find by images of specific person.

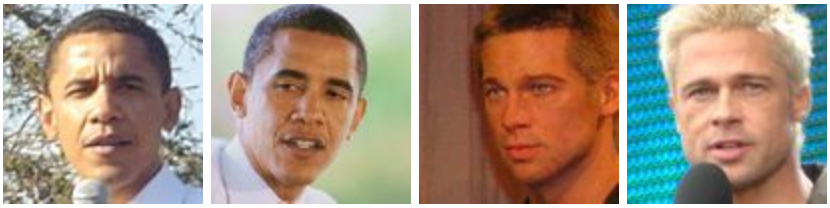
The task of face recognition can be very challenger. The person who appears in the image may be in different poses, with different light conditions, with eyeglasses. Because this, extract the important features of a face can be very difficult.

For this project, I'm proposing to build a model using a Convolutional Neural Network (CNN) for the task of face recognition. A model that uses a CNN, allow us to extract a wide range of unique features from an image and we can use this features to compare a face with others.

To solve this problem, my proposal is to construct a template that receives as input an image, add labels with the name of the person that appears in the image.

Datasets and Inputs

I'm going to use a version of PubFig[3] dataset from Columbia University. This dataset is called PubFig83[4]. This dataset has a 8300 cropped facial images, made up for each 83 public figures.



Samples of the dataset

The idea behind to use this dataset is to take advantage of the work did by the dataset authors to remove duplicates and resize the pictures. The dataset will be divided as follows:

80% of data for training set

10% of data for validation set

10% of data for test set

Solution Statement

For this project I'm planning to use Deep Learning with a Convolutional Neural Network to recognize people in a image. A CNN model may be more effective to extract important features in a given image. The basic idea is:

- Receive an image as input
- Detect faces contained in the image
- Crop this faces and converts to the gray scales
- Use the cropped image in the gray scale in the CNN model
- Get the predicted label from the model and draw a box around the identified person

Benchmark Model

I plan to compare the results of my CNN model for face recognition versus the a OpenCV EigenFaces Recognizer. The plan is to compare accuracy of these two models and check what perform better. For doing this, I'm planning to train the same subset of data in the CNN model and OpenCV EigenFaces Recognizer, after that, check the accuracy comparing the two models.

Another interesting experiment I'm planning to do, is to compare accuracy with a simpler model like Logistic Regression. It will be useful to create a baseline and check how proposed model compare with this.

Evaluation Metrics

To compare the models, I'm planning to use accuracy. I think it's a simple and straightforward metric, basically it's a fraction of how many labels the model has correctly identified. During the training phase, I will use Categorical Cross-Entropy [5] as the Loss Function. A loss function is important during training because it makes

possible to understand whether or not the model is improving during the process. The important point here is to try to minimize the loss.

Project Design

The final goal of this project is to create a model where it is possible to do the recognition of faces in a image. An important step to the success of this model is the data collect to train the model. As I said earlier, the dataset used is a subset of the PubFig [3].

Once the dataset has been collected, it will be divided into 3 subsets: Training, Testing and Validation.

The model will receive an image as input. Before start the process of face recognition using this image, will be necessary take some steps to start the process:

1 - Find all faces in the images

The first thing to do is look to the image and check if it contains faces. If it does not contains faces, the model will alert that there is no faces in the image. For this project, I will use a library called Face Recognition[6] and Dlib[7] to the detection task. The advantage of this Library is that it uses a HOG[8] algorithm in the process of detection.

The image above, shows the face detected and the HOG algorithm applied to the face.

Sometimes the process of detection can return false positives (it says there is a face, but does not have faces in the image). To minimize this effect, the model will use a threshold size to ignore detections with size greater than the threshold.

2 - Crop the faces

Crop the faces from the original image is an important step after the detection, because it remove all the unimportant data and keep just the face to apply the model. The faces will be cropped and resized to stay at same size. Beside that, the model will ignore colors and just convert the image to a Gray scale.

Barack Obama's face in grayscale

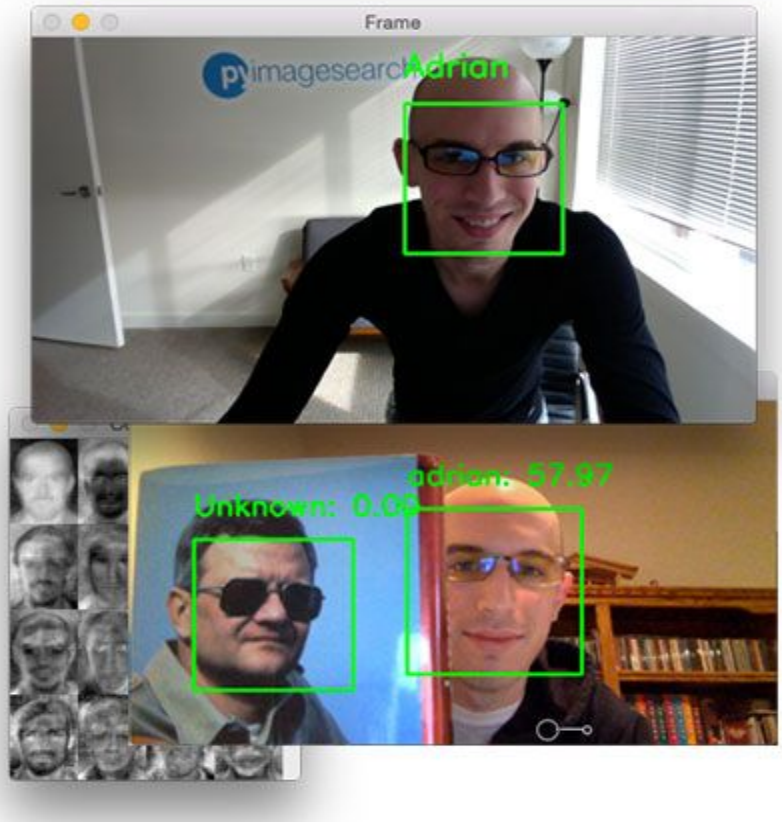
My hypothesis to use Gray scale is to remove noise, but I'm not sure if it will work as expected. Because this, I will try to use color images too and check the accuracy of the two approaches.

3 - Recognize Faces

I'm planning to use Deep Learning with a Convolutional Neural Network (CNN) model to this task. My plan is to test 3 different architectures: a custom model, a VGG16 [9] and a ResNet50 [10]. The model that will provide the best accuracy will be part of the final model. The idea is to change some params and check the accuracy. For example, change the filter size, stride, number of layers, introduce pooling layers and dropout layers. I will use the Keras Library to build the model and the network architecture.

4 - Identify the person

Once the face was recognized, it is necessary to get the correct label. Besides the label will be show the associated probability of the prediction. With face coordinates, I will use OpenCV to draw a box around the face with the label and the probability. To faces that was not recognized, a box with' the Unknown label will be draw into the image.



The image show the output of the model [11]

The model will be presented in a Jupyter notebook.

References

[1] Paul Viola, Michael J Jones: International Journal of Computer Vision 57, pp. 137-154, Netherlands, 2004.

[2] The Verge article about Pornohub, https://www.theverge.com/2017/10/11/16459646/pornhub-machine-learning-ai-video-tagging
[3] PubFig dataset page, http://www.cs.columbia.edu/CAVE/databases/pubfig/download/
[4] PubFig83 dataset page, http://vision.seas.harvard.edu/pubfig83/
[5] Keras documentation about Categorical Cross Entropy, https://keras.io/losses/#categorical_crossentropy
[6] Face Recognition source code, https://github.com/ageitgey/face_recognition#face-recognition
[7] Dlib page, http://dlib.net/
[8] Navneet Dalal, Bill Triggs. Histogram of Oriented Gradients for Human Detection, 2005 http://lear.inrialpes.fr/people/triggs/pubs/Dalal-cvpr05.pdf
[9] K. Simonyan, A. Zisserman. A very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv: 1409.1556
[10] Kaiming He, Xiangyu Zhang, Shaoging Ren, Jian Sun. A Deep Residual Learning for Image Recognition. arXiv:1512.03385
[11] image from https://www.pyimagesearch.com/pyimagesearch-gurus/
[12]https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78
[13]Nicolas Pinto, Zak Stone, Todd Zickler, and David D. Cox, "Scaling Up Biologically-Inspired Computer Vision: A Case Study in Unconstrained Face Recognition on Facebook", *Proc. Workshop on Biologically Consistent Vision*