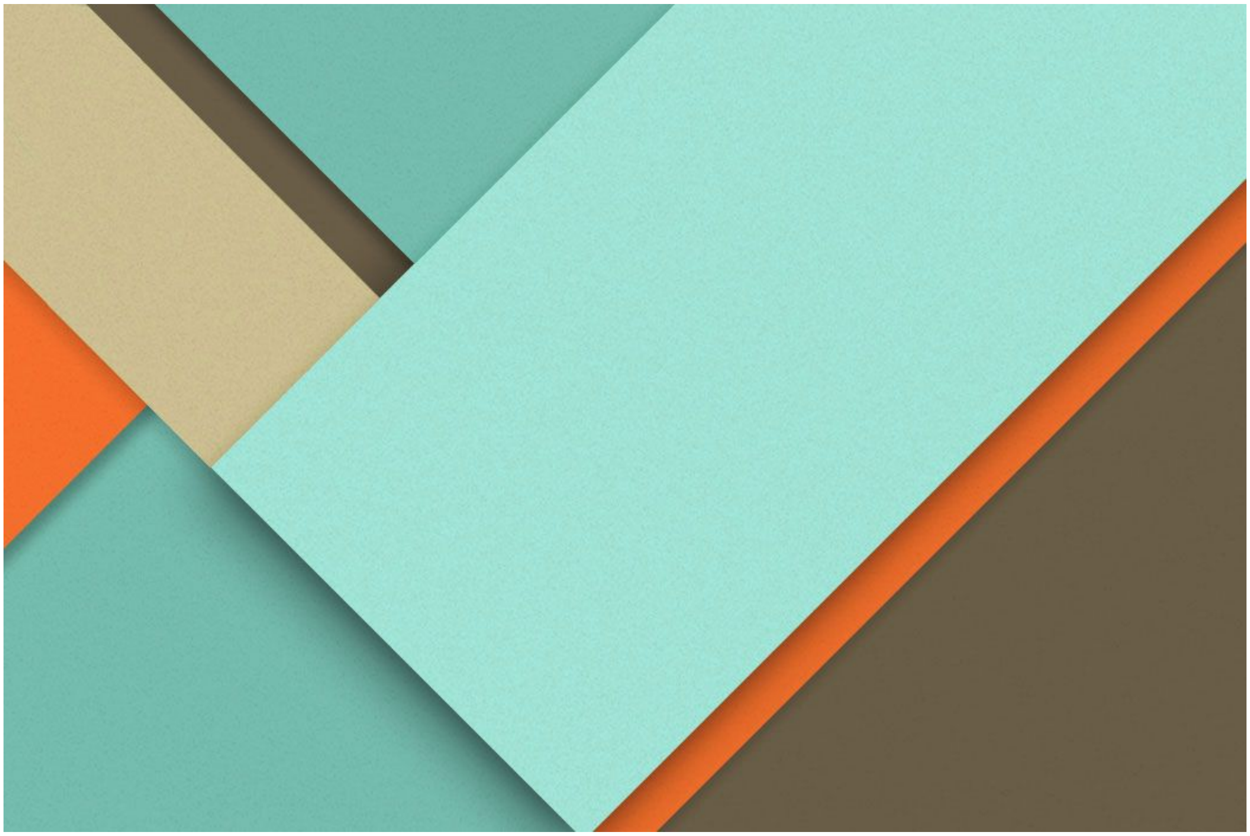

Facial Keypoints Detection

Machine Learning Engineer Nanodegree

Capstone Proposal

10.01.2017

Amil Khare
Jaypee Institute Of Information Technology
Sector - 62 , A - 10
Noida, U.P.
India. 201307

Domain Background

The Capstone Project that I have chosen is the Facial Keypoints Detection. I chose the medical field , since I am deeply interested in using Machine Learning in Medicine due to the vast amount of data, research available and so many results to be found. I had in my mind to devote my time and skills in medicine but was confused about the process of getting data. Hence I turned my mind to kaggle and I thought what better way to even get my results judged compared to others other than a past competition. So after searching for a bit this Competition got my attention.

We know that there is technology that can judge our results like advanced machines , but we could use Machine Learning to speed up the process and compare our results to those advanced machine and try to make our model work better and better for such applications.

Problem Statement

The objective of this task is to predict keypoint positions on face images. This can be used as a building block in several applications, such as:

- tracking faces in images and video
- analysing facial expressions
- detecting dysmorphic facial signs for medical diagnosis
- biometrics / face recognition

Detecting facial keypoints is a very challenging problem. By keypoints I mean that for a person the position of important and common features such as eyes , nose , upper lip , lower lip etc. Facial features vary greatly from one individual to another, and even for a single individual, there is a large amount of variation due to 3D pose, size, position, viewing angle, and illumination conditions. Computer vision research has come a long way in addressing these difficulties, but there remain many opportunities for improvement.

Data sets & Inputs

The data set was easily obtained from the kaggle competition. In the following lines I will be citing the information as obtained from kaggle.

The data set for this competition was graciously provided by Dr. Yoshua Bengio of the University of Montreal.

Each predicted keypoint is specified by an (x,y) real-valued pair in the space of pixel indices. There are 15 key points, which represent the following elements of the face:

left_eye_center, right_eye_center, left_eye_inner_corner, left_eye_outer_corner, right_eye_inner_corner, right_eye_outer_corner, left_eyebrow_inner_end, left_eyebrow_outer_end, right_eyebrow_inner_end, right_eyebrow_outer_end, nose_tip, mouth_left_corner, mouth_right_corner, mouth_center_top_lip, mouth_center_bottom_lip

Left and right here refers to the point of view of the subject.

In some examples, some of the target keypoint positions are missing (encoded as missing entries in the csv, i.e., with nothing between two commas).

The input image is given in the last field of the data files, and consists of a list of pixels (ordered by row), as integers in (0,255). The images are 96x96 pixels.

Data files

- **training.csv:** list of training 7049 images. Each row contains the (x,y) coordinates for 15 keypoints, and image data as row-ordered list of pixels.
- **test.csv:** list of 1783 test images. Each row contains ImageId and image data as row-ordered list of pixels
- **submissionFileFormat.csv:** list of 27124 keypoints to predict. Each row contains a RowId, ImageId, FeatureName, Location. FeatureName are "left_eye_center_x," "right_eyebrow_outer_end_y," etc. Location is what you need to predict.

Solution Statements

A person's facial features has some distinctive looks/keypoints that differentiate it from it's face and other features. Let's take Nose for example we know there has to be two nostrils in a nose and we can further make out the structure by seeing the bump of the bone of the nose in the face. Similarly let's take eyes , we know that eyes have eyelashes and above have eyebrows. Although eyebrows won't help much since their position varies greatly with respect to eyes however eyelashes do help and similarly we know that eyes have Sclera (the white part). This too helps in distinguishing the inner part of eye from the outer part of eye.

Using such similar features we can actually try to detect the facial keypoints in a person's face. I similarly believe that taking multiple images of a person in different angles too might help in this since it will not only prevent the wrong detection but also improve our accuracy.

I believe that there are 2 approaches to this problem that I would like to apply. One with PCA and other without PCA. Of course through enough research on the internet I have come to the conclusion that Neural Networks will be the best option since , initially we don't know how the images are taken. They will provide with a good/accurate model as compared to Random Forest, SVM , etc. Since each person might have varied position of keypoints with respect to others , this might be possible with model such as neural nets since the weight and threshold will certainly help. I believe that due to uncertainty in the data and facial keypoints neural nets is the best model.

As for PCA it might be the case that PCA brings out much more information than something that's already available and can further improve accuracy , since I have had quite good experience with PCA in facial recognition.

Benchmark Model

I believe that after observing the leaderboard my initial task/ objective is to achieve a rank in the top 100 of the kaggle leaderboard. However at first I would like a model that satisfies my intentions. I am almost sure some models will not be good , hence I would like to implement some of the models other than Neural Network and see their results compared with my other models , because , of course I got a a lot to learn and it might be the case that ' TOP 100 ' is tough to achieve , considering it's already a completed competition.

Evaluation Metrics

Root Mean Squared Error (RMSE)

The evaluation metric I chose is the root mean squared error. RMSE is very common and is a suitable general-purpose error metric. Compared to the Mean Absolute Error, RMSE punishes large errors

Moreover submission made to the competition , i.e. the kaggle platform will also help in evaluation of the model compared to others.

Project Design

Here I will be discussing the full algorithm/model in detail.

Let's firstly begin with data. There is a lot of chance that data cleaning and data analysis is required for checking missing values or noises or features that give less information. After this I will start building a Neural Network. I have heard quite a lot about Theano & Lasagne and have thought to give it a try for this problem.

I too found out about Lenet-5 which is groundbreaking in Computer Vision and plan to use it too.

One big problem that might occur is overfitting. An overfitting net can generally be made to perform better by using more data.

Moreover to make it better it's then about tuning the parameters where GridSearch might help a lot. However even after such work things don't turn out good , I plan to try other models.