



Facial Keypoints Detection

Karim Mohamed El_gohary

November 20th, 2019

Domain Background

Detecting facial keypoints is a very challenging problem. 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. The facial keypoints detection in several applications as (tracking faces in images and video, biometrics / face recognition)

Problem Statement

The problem is prediction problem, The objective of this task is to predict keypoint positions on face images.

Datasets and Inputs

Source:- the dataset from competition on kaggle this is the [link](#)

Data Description:-

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.

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

Solution Statement

The first thing in the solution is to explore the dataset and preprocessing/cleaning the data, and prepare the data into X_train, y_train then we will build the model the model here is CNN with 13 layers and (3*3) filter size and pool_size = (2*2), with activation function 'relu', the optimizer is 'adam'

Benchmark Model

I will use private leaderboard as reference to compare my result to them

Evaluation Metrics

As we have to predict new values. Our results evaluated on the basics of 'mae' (mean absolute error) .

The loss function will be 'mean_squared_error' and the optimizer 'adam'

Project Design

1) import dependencies(numpy,pandas,matplotlib,keras)

2) Explore the dataset

3) Preprocessing/cleaning data set :

- Filling the missing value with previous values in that row.

- separate the labels and features.

- reshape and convert the image into float value.

4) define our model:-

I am using keras and simple dense layers. For loss function I am using 'mse' (mean squared error) as we have to predict new values. Our results evaluated on the basics of 'mae' (mean absolute error), i will create 13 layer .