Novaturient Task 1

Categorical Smile Detection

Chaitanya Kini

Problem: The task at hand is to detect from a given face whether the person is smiling or no and if he/she is smiling, the extent of the smile had to be classified between zero and hundred.

Obstacles At Hand: This very problem is achievable to an extent where any smile on a persons can be detected with a very high accuracy and that smile can be successfully categorised. But, the major setback here that was faced during this attempt was the major lack of data. Training a successful model required a lot of data and this was not available at the time. Given the amount of training data I could find there was a high chance of overfitting which eventually is nothing but a failed model.

Proposed Solution: The initial solution to this problem was to train an CNN network followed by densely connected flattened layers that helped us detect whether it was a smile or no.

Classification:

The model's last layer involved a softmax activation that gave us a value between 0 and 1 stating the probability of the smile. Furthermore, if the output was >0.5, the input image classified as a smile whereas if the output <0.5 , the input image did not have a smile.

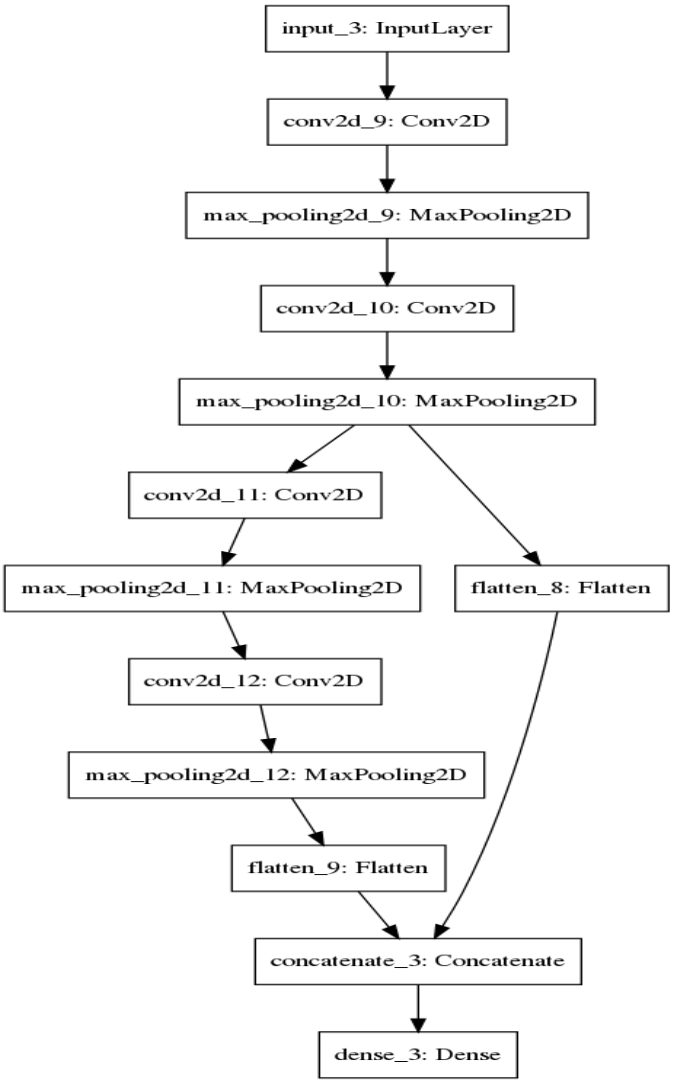
Categorical Classification:

Furthermore, once we have the output and provided we have the output to be greater
than 0.5, we map that value between 0 and 100 and give a categorical output.

For Ex. If the output is 0.78 then the categorical output would be (0.78-0.5)*2 = 0.56.

An important question here is, how will we get odd categorical values? The solution to this is, limiting the ouput softmax value to 3 decimal values, which can then be used for rounding up.

Proposed CNN model:



Solution Given:

Since I wasn't able to gather data, I have come up with a solution which includes Haar Cascades along with opencv libraries to track a live smile.

The flow plan is as follows:

1. Feeding the input video/webcam input.
2. Detecting the faces in the video using Haar Cascade(haarcascade_frontalface_default.xml)

3. Detecting the smile in the face using Haar Cascade(haarcascade_smile.xml)
4. Monitoring the ratio of the width of the smile's bounding box to the width of the faces's bounding box and converting this ratio to a categorical value between 0 and 100.

Conclusion:

Because of lack of data to train the smile detection, I deemed it sufficient to develop an image processing algorithm using opencv to solve the problem.

Execution: To run the program, run the smile.py file and make sure the video under consideration is in the same folder. Change the video name in the 10th line of the file and run it to see the results.