Task-1 Report: Face Detection

Soumith Reddy Chinthalapally

soumithr@buffalo.edu (50336650)

Implementation:

I have used **cv2.CascadeClassifier** for face detection using Haar cascades approach.

Face detection using Haar cascades is a machine learning based approach where a cascade function is trained with a set of input data. OpenCV already contains many pre-trained classifiers for face, eyes, smiles, etc. To load this model for face I have downloaded "**haarcascade_frontalface_default.xml**" file.

I have used **cv2.CascadeClassifier.detectMultiScale()** to find faces. The parameters sent to this model are

scaleFactor: Parameter specifying how much the image size is reduced at each image scale. This scale factor is used to create scale pyramid. I am using a scale factor equal to 1. 05 in the code

minNeighbors: Parameter specifying how many neighbors each candidate rectangle should have to retain it. This parameter will affect the quality of the detected faces: higher value results in less detections but with higher quality. I am using 5 in the code.

We store the above result in faces. **Faces** contains a list of coordinates for the rectangular regions where faces were found. We use these coordinates to draw the rectangles in our image.

Faces contains 4 values stored in x, y, w, h. x, y indicates the co-ordinates of top left corner of face and w, h indicates the width and height of the detected face. We draw a rectangle using these values and using cv2.Rectangle function to show it on the image. We append these values to a JSON file and store the JSON file.

Results and Implementation Issues:

I have achieved a F1 score of 0.773 on the training data.

The model is fast and detects faces accurately in most of the cases.

The implementation is simple, but the parameters tuning is a bit difficult. I have tried different values for scaleFactor and minNeighbours to increase F1 score. After some trials found that using minNeighbours value equals to 3 or 4 or 5 gives optimal results based on the data. I have used 4 for our data which gave optimal F1 value. scaleFactor value anyway between 1.03-1.06 is good for a good detection model. I have used 1.05 for my model.

Disadvantages:

One of the issue I observed from the detected faces is that it is unable to detect faces which are not front facing. This can be seen from the below generated image clearly.