Human Face Detection using Resnet10_SSD.

Group ID :- PR_4

Name: Esha Gawate **Roll No:** - 34
Name: Gauri Khadatkar **Roll No: -** 53

Abstract

Human face detection has become a major field of interest in current research. Further, the algorithms that exist are very much specific to the kind of images they would take as input and detect faces. The problem is to detect faces in the given image which is captured in real-time. Face detection is performed on live acquired images without any application field in mind. To detect faces we use Resnet-101_SSD until we get the confidence of the detected faces.

Introduction

With the advent of new technologies, there is a need for image analysis. Face detection may feel trivial for any human but for a machine, it is very tedious to analyze if an image contains a human face or not. On the basis of facial features, expressions, and orientation, the machine decides it. The detection of a face is very complex because of lighting effects, the presence of any object over the face such as sunglasses, and image resolution. Face detection is a special type of Object Detection in which facial features are extracted. An image is manually captured by a real-time camera and then processed and the human faces are detected. The main aim of Face detection is to determine whether there are faces in the image or not. Face detection is an essential requirement for all the facial analysis algorithms such as face alignment, face modeling, face relighting, face recognition, face verification, face authentication, head pose tracking, facial expression tracking, gender and age recognition, and many more. The input to our algorithm is an image that would be captured manually by the user. We then use a ResNet-10 Architecture to output the predicted faces.

Related work

A face detection framework is designed where a new representation of the image is called an ‘integral image’ for faster computations. It uses a simple and efficient classifier in a “cascade” which allows the background of the image to be quickly discarded while spending more computation on promising face-like regions[1]. The demonstration of state-of-the-art face detection performance on two benchmark datasets using the Faster R-CNN. Experimental results suggest that its effectiveness comes from the region proposal network (RPN) module. Due to the sharing of convolutional layers between the RPN and Fast R-CNN detector module, it is possible to use a deep CNN in RPN without extra computational[2]. A novel detection network named Dual Shot Face Detector(DSFD) which inherits the architecture of SSD and introduces a Feature Enhance Module (FEM). It transferred the original feature maps to extend the single shot detector to a dual shot detector used for detecting faces in an image[3]. Face detection can be used as a general object detection task. There is an implementation of face detection based on the YOLOv5 object detector called YOLO5FaceDetection[4].

Dataset and Features:

The usage of the deep learning pre-trained model to detect the faces from the image from different angles, and the model which is used is the Caffe model “res101_300 x 300_ssd_iter_140000.caffemodel”. In this model, we will provide one image as input or we can take a real-time image from a webcam for the identification of multiple faces. This model has weights that have been learned from lots of different images. This is the pre-trained model which can be directly used. There is no need to train it from scratch using the pretrained file provided by the SSD algorithm. This pretrained file works on an image resolution of 300 x 300. This deploy.prototxt.txt file is the architecture of the Single Shot Detection (SSD) Model which has a backbone of Resnet. Caffe (Convolutional Architecture for Fast Feature Embedding) is a deep learning framework that allows users to create image classification and image segmentation models. Initially, users create and save their models as plain text .prototxt files. A list of the neural network layers in a model contains each layer's parameters, including its name, type, input dimensions, and output dimensions. It also contains the specifications for connections between layers. Below are the images obtained after implementation of the Face Detection Algorithm.

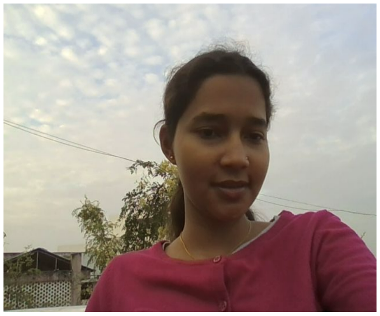


Fig 1. Input image

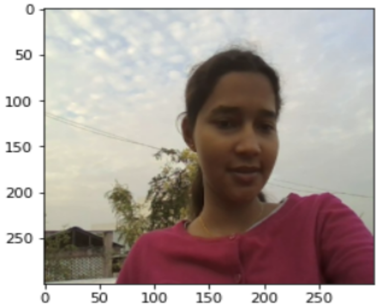
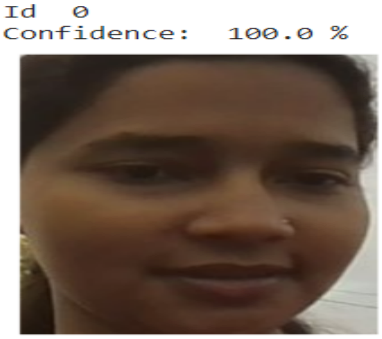


Fig 2. Resized Image



```
Id  0
Confidence:  100.0 %
```

Fig.3. Confidence of detecting a human face

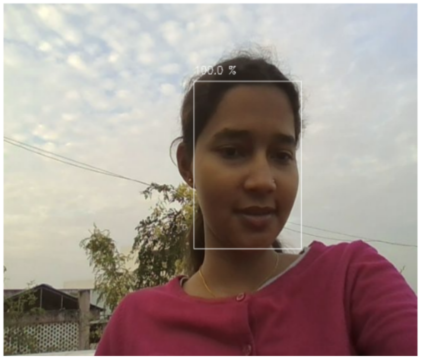


Fig 4. Face Detection

<u>Methods:</u>

Firstly, a photo is captured by the camera whose resolution is resized to 300 x 300. The aspect ratio is then calculated by considering the ratio of the x-axis of the original image to its resized image. The same is done for the y-axis as well. The resized image is facilitated towards the preprocessing. cv2.dnn.blobFromImage performs Mean Subtraction and scales the image. It is used to help combat illumination changes in the input images. The image is then sent to perform the operations by adjusting the weights and detecting whether it contains any face of a human. These detections also predict how much confidence it is predicting that a human face is detected and also give us the face's alignment. The threshold by which it will predict whether a face is detected or not is when the value of confidence is greater than 0.90.

Experiments/Results/Discussion:

The face detection process is done by using the Residual SSD Network which consists of and residual network due to which it will be able to carry out the features which are in the previous layers and it will be given to the fully connected neural network so that it can extract the exact feature for increasing the accuracy.

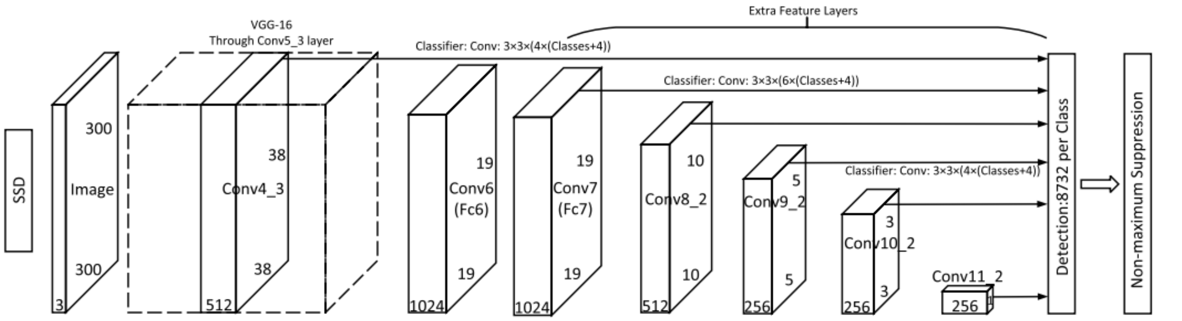


Fig. 5. The Network Structure of SSD.

We have used the improvised learning algorithm of SSD in which a Residual network is introduced. Because of the disadvantages in the traditional networks here basically the VGG16 model is interchanged by the deep neural networks for better extraction ability. If the features and the residual networks transfer the optimal weights from the front layer to fully connected layers it solves the problem of vanishing gradient and will reduce the rate of false detection and use the Relu function which makes the learning cycle short and reduces the time. For this, we have the pre-trained file of Resnet10 SSD and the deploy prototxt.txt

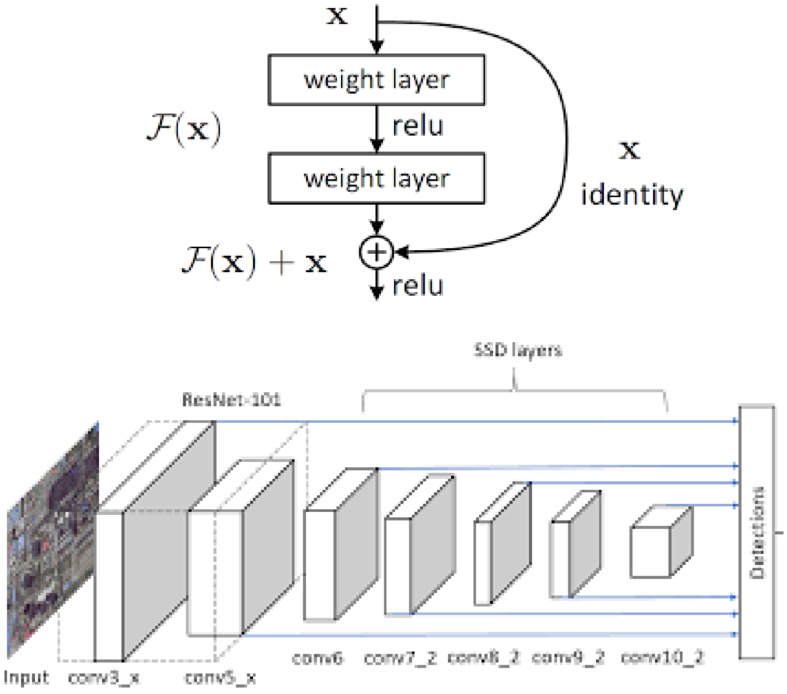


Fig. 6. ResNet_101+SSd Network.

Conclusion/Future Work: -

Due to recent advancement in the field of Face recognition systems there is a need for processing applications. The significance of this project as a research area is increasing rapidly. Implementations of the system are crime prevention, video surveillance, person verification and similar security activities. Resnet-101+ SSD Algorithm is used for face detection. This project is basically about face detection only. But our upcoming work will include face recognition, and marking the attendance of the student in a particular excel sheet. The work of this project will be to provide the dataset of our pictures to the model and train it for our images to recognize the person correctly and mark the attendance in the excel sheet against the name of that particular person.

Contribution: -

The topic was researched. There is an attempt to interface the model with real-time access and real-time face detection. Decoding the model of the deploy prototype.txt and understanding the Caffe model were done. Apart from it, the analysis of the code and the algorithms involved were studied.

References/Bibliography

[1] PAUL VIOLA, MICHAEL J. JONES.Robust Real-Time Face Detection. International Journal of Computer Vision, 2004

[2] Huaizu Jiang, Erik Learned-Miller. Face Detection with the Faster R-CNN , 2016.

[3] Li, Jian, et al. "DSFD: dual shot face detector." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019.

[4] Qi, Delong, et al. "YOLO5Face: why reinventing a face detector." *arXiv preprint arXiv:2105.12931* (2021)

https://medium.com/featurepreneur/object-detection-using-single-shot-multibox-detection-ssd-and-opencvs-deep-neural-network-dnn-d983e9d52652.

Source of the ‘prototxt.txt’ file: -
https://github.com/gopinath-balu/computer_vision/blob/master/CAFFE_DNN/res10_300x300_ssd_iter_140000.caffemodel