Facial detection using deep learning

IOP Conf. Series: Materials Science and Engineering **263** (2017) 042092 doi:10.1088/1757-899X/263/4/042092

Facial detection using deep learning

Manik Sharma, J Anuradha, H KManne and G S CKashyap
School of Computing Science and Engineering, VITUniversity,Vellore - 632014, India

E-mail:januradha@vit.ac.in

Abstract. In the recent past, we have observed that Facebook has developed an uncanny ability to recognize people in photographs. Previously, we had to tag people in photos by clicking on them and typing their name. Now as soon as we upload a photo, Facebook tags everyone on its own. Facebook can recognize faces with 98% accuracy which is pretty much as good as humans can do. This technology is called Face Detection. Face detection is a popular topic in biometrics. We have surveillance cameras in public places for video capture as well as security purposes. The main advantages of this algorithm over other are uniqueness and approval. We need speed and accuracy to identify. But face detection is really a series of several related problems: First, look at a picture and find all the faces in it. Second, focus on each face and understand that even if a face is turned in a weird direction or in bad lighting, it is still the same person. Third select features which can be used to identify each face uniquely like size of the eyes, face etc. Finally, compare these features to data we have to find the person name. As a human, your brain is wired to do all of this automatically and instantly. In fact, humans are too good at recognizing faces. Computers are not capable of this kind of high-level generalization, so we must teach them how to do each step in this process separately. The growth of face detection is largely driven by growing applications such as credit card verification, surveillance video images, authentication for banking and security system access.

1. Introduction

In the last few decades, a lot of research has been done in face detection. We can identify a person's face without the help of any human support. In this paper, a system is implemented to evaluate the human face detection.

A model that can be used in many types of devices to detect digital images is called face detection. It is a specific case of object-class detection. Isearches for the location and size of all the features belongingto a given class. The main priority of this model will be on the frontal face detection. In this face-detection model, it firstly detects expected human eye areasby first evaluating all the possible valley regions in given gray-level frame or picture. The fitness value of a feature is determined based on its projection on the eigen-faces. After so many iterations, the symmetryof the face is determined and the presence of various facial features is tested and confirmed. In other terminologies, we call facedetection as face-priority AF (auto focus), a function that detects human faces so that focus will be set along with appropriate exposure.OpenCV will be used for face detection.

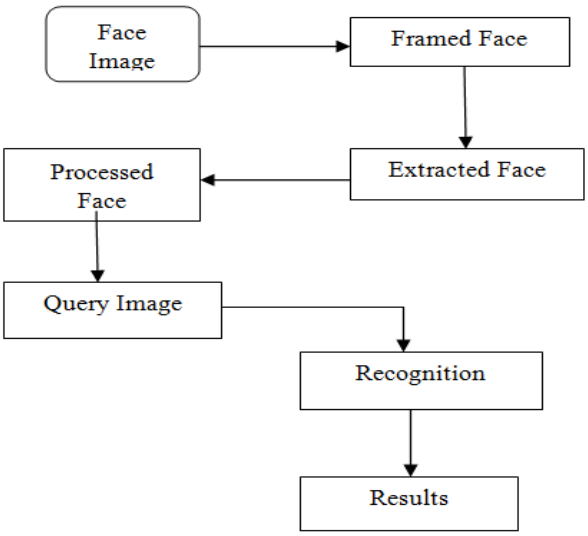

Figure.1 Flow Chart of Face Detection Model

1.1OpenCV

OpenCV (Open Source Computer Vision) is a library started by Intel in 1999. It is a classifier that helps in processing the images. The main focus is mainly on image processing and can be implemented on the latest algorithms. It comes with a programming interface to Python. It is used under a BSD license to be used in academic projects. Itcomes with the FaceDetection class for face detection. To perform the face detection, the following modules need to be imported.

1. cv2 – It is an OpenCVpackage object or module and has the functions image processing.
2. numpy – The image will be stored in numpy arrays.
3. sys–to perform console related functions like taking input from theconsole.

The first step will be detecting the face in each image and use the HaarCascadewhich is given by OpenCV. The haarcascadesare present inOpenCV and can be found in the location /data/haarcascades> directory of theOpenCV installation. the haarcascade_frontalface_default.xml and haarcascsade_eye.xml are used to detect the face and eyes respectively. The xml files will be loaded using cv2.CascadeClasssifier function which takes the path to it.

1.2 Haarcascades_frontalface_default

Using Haarcascade classifier, the face detection is done by the following classifiers: haarcascade_eye and haarcascade_frontalface_default.To train the classifier, the algorithm requires a number of images (positive and negative). The features are extracted from the images. In the below figure 1, the use of haar feature is shown. Every feature considered by this model is a single value gathered by getting the difference between the aggregate of pixels which come under white rectangle and the aggregate of the pixels which come under the black rectangle.

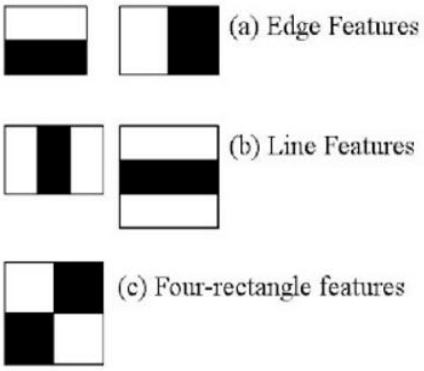

Figure. 2 HaarCascades_frontalfaces_default working

1.3Deep Belief Network (DBN)

In machine learning, DBN is described as a graphical model or type of deep learning architecture. The Deep Belief Network probabilistically reconstructs its inputs. Below is the schematic overview of DBN. The arrows are the directed connections present in the graphical model. It is a composition of simple and unsupervised networks such as Boltzmann machines and autoencoders, where each sub-networks hidden layer serves as the visible layer for the next.

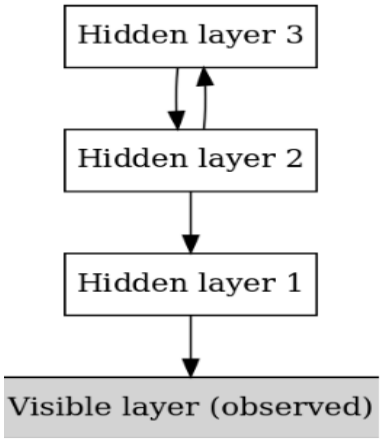

Figure.3 DBN architecture

2 Literature survey

Face detection is a great study in the computer vision. In olden days i.e.(before 2000) many studies and practical performances of the face detection was not satisfactory until Viola and Jones proposed a work. These Viola and Jones [1][2] are the first who are applying rectangular boxes for the face. But, it has lot of drawbacks as its feature size was large. In a 24 × 24 image, the total number of Haar_like features is 160,000[3] and also it is not handled for wild faces and frontal faces.

After finding the above problem, people have put lot of effort to introduce with more complicated features (HOG, SIFT, SURF, and ACF)[4][5]. In [6] introduced the new feature NPD which will differentiate the two pixels intensity. Another well known method is Dlib[7] which took support vector machine as a classifier in the face detection.

Enlarging the robustness in the detection is another greatly studied topic. One of the simple method is to combine multiple detections that should be trained separately in different views [8] .Zhu et al. [9] is applied multiple de-formable models to take faces in different views.Shenet[10] al given a

retrieval based model combined with different learning. These models require training and testing where it will take more time and its performance is less. In 2002 Garcia et al. [11] introduced a neural network to find the semi -frontal human faces in the complex images. In 2005 Osadchy et al. [12] trained a convolutional neural network for the face detection.

Over the last few years, a lot of face detection and face recognition work has been done. As it is the best way for recognizing a person. Because it does not require any human work to recognize faces. Since a lot of methods invented for face recognition and face detection.

3. Methodology

DBN is an implementation of deep neural net. These DBN create a neural network layer by layer i.e., starting from the input layer to end with output layer. In between input and output, different no of hidden layers are present. Every layer should be trained with RBM. An RBM is an extract feature to re-construct inputs. By combining all RBM's we are introducing a collaborating method getting a power full new model which solves our problem that is DBN. Just like MLP, DBN is also considered a back propagation with RBM to provide a powerful trained network.

In terms of network structure, a DBN is identical to MLP. But when we come to training, those are entirely different. This training process is the key feature for developing powerful trained networks. A DBN is called as a stack of RBN's. In these hidden layers one RBN is visible layer for the above one. The first RBM constructed as input is possible after that hidden layer of first RBN has visible layers for the second RBN. If the second RBN is trained with the help of first, this process is repeated until every layer in the network finishes the training process. In the face detection process, hidden layer finds the edges of the face and visible layer finds the features of the faces. A DBN can work globally which effects a model should improve performance. Like our camera lenses slowly focus on the picture the reason DBN works better is highly technical and a stack RBM will work as a single unit. After the initial training these models create RBMs which detect the hidden patterns in the dialogue. To finish the training we introduce labels to the pattern for the supervised learning and need sample set of features which will increase our results in better way. For this, we need a small set of data set which should be reasonable for our real world application. The main reason we are going for DBN is because the training process is completed in reasonable manner. It provides very good results compared to other algorithms. In DBN we are using recurrent feed forward neural network. The reason we are going for recurrent feed forward neural network is it can handle change in input /output size. In conventional we can handle fixed no of input /output size. In this method (RFFN) the output layer will forward the input for next layer after its going to end. If the training model is not complete, then it comes back to start and train by considering previous output training to input for present training.

$$P(x, h^1, \ldots, h^\ell) = \left(\prod_{k=0}^{\ell-2} P(h^k | h^{k+1}) \right) P(h^{\ell-1}, h^\ell)$$

Here, x is the observed vector,

ℓ is the no of hidden layers,

h^k is represent each hidden layer,

$P(h^{\ell-1}, h^\ell)$ is visible layer and hidden layer joint for the RBM at level k,

$P(h^{k-1}|h^k)$ at $x = h^0$ is the conditional distribution for visible units.

3.1 Algorithm

Step 1: Train the 1st layer with RBM at raw input $x = h^0$ visible units.

Step 2: By using 1st layer we will get visible pattern data for the second layer.

Step 3: Train the second layer from output 1st layer as visible layer and with combination of2nd hidden layer.

Step 4: Repeat 2, 3 steps until it completes the total training process.

Step 5: Tune all the parameters by using a small set of dataset to increase the accuracy of results i.e., supervised training creation.

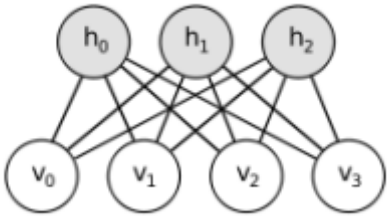

Fig 4: RBM connections between visible and hidden layers

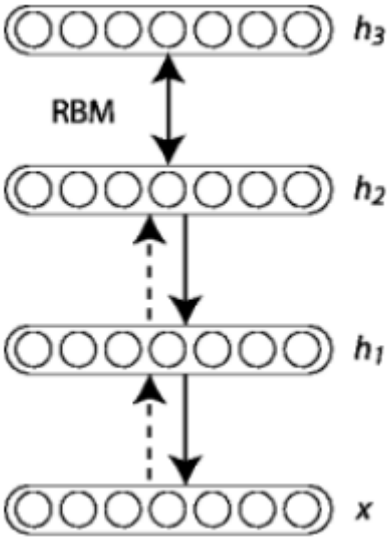

Figure. 5 A simple graph image for DBN with 3 hidden layers

4. Implementationand Results

First, we need to install OpenCV package for python. This package can be downloaded from python website or by using *pip install* command.

Then we use two cascade files *haarcascade_frontalface_default.xml* and *haarcascade_eye.*xml files which are available under GNU licence and can be used without permission also. The whole code is implemented in python and need a working webcam to capture images or videos.The model is implemented successfully and is able to recognise faces in still images,

videos, paintings and webcam captures. For still images and paintings the model is able to recognise eyes also.Result for different images is shown below.

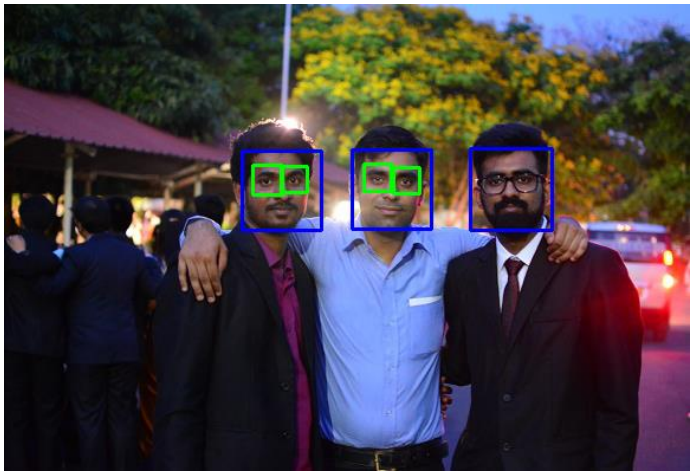

Figure. 6 For still image

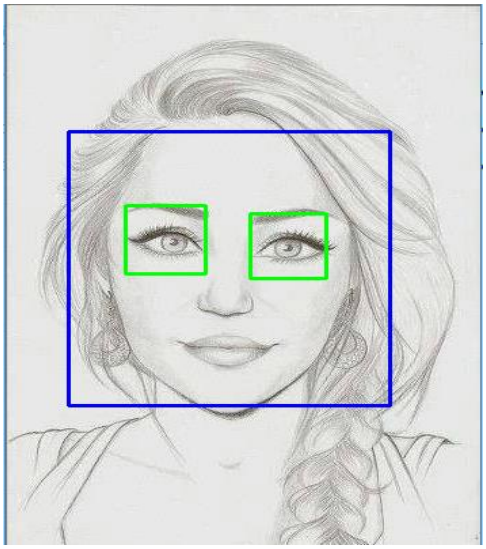

Figure. 7: Hand Drawn Image

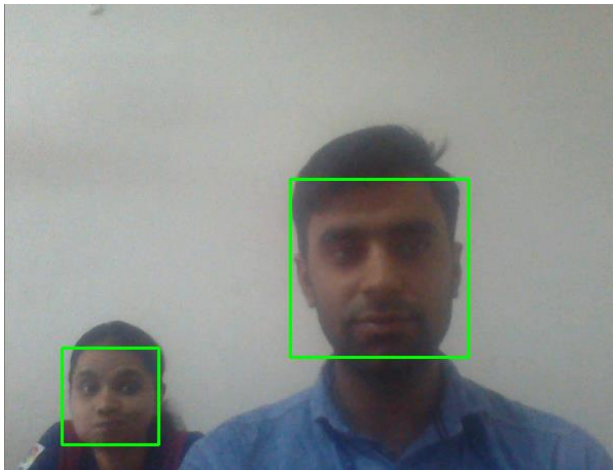

Figure. 8: Webcam Image

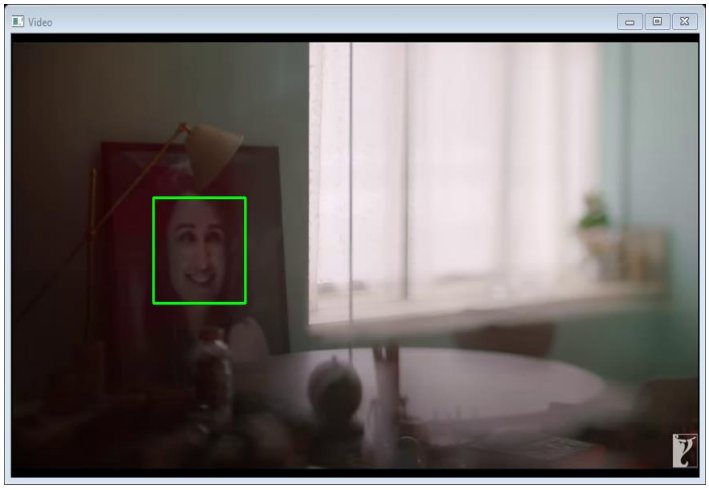

Figure. 9: Blurred Image

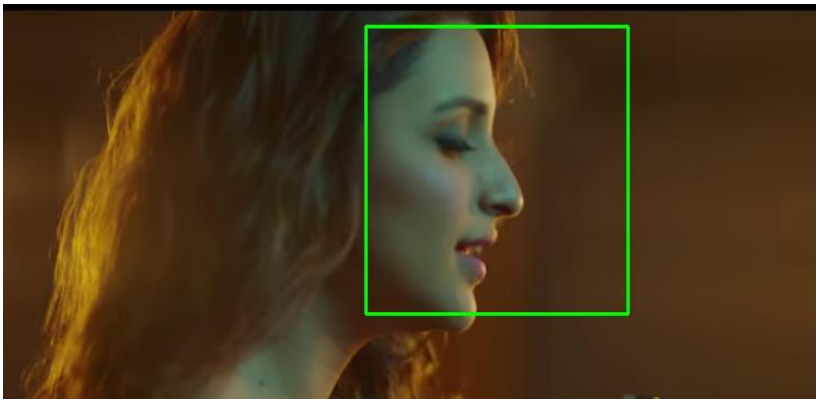

Figure. 10: Side Face Image

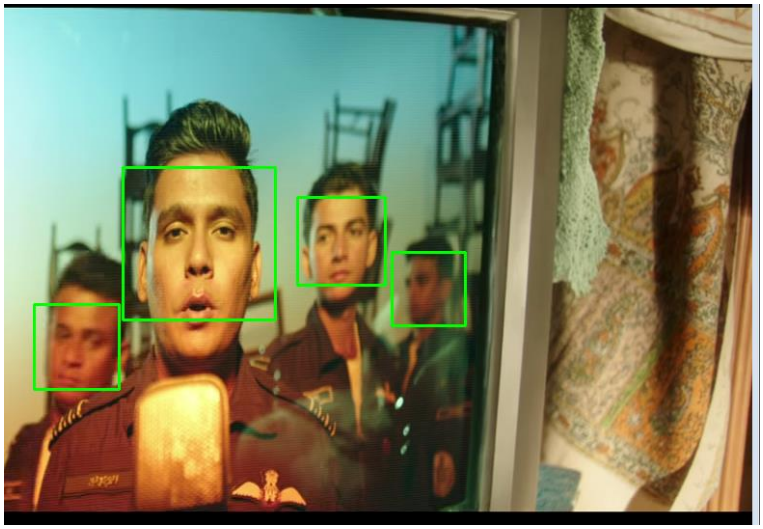

Figure. 11: Video Result

5. Conclusions

The proposed model is able to recognize faces correctly but when tried for videos, it takes more time for processing. The advantage of this model is that it is able to recognize blurred images and side face images also which other traditional models are incapable of recognizing in such case. The onlydrawback is that it fails to recognize eyes with glasses. In future, this can be extended to recognize persons using video capture which will be helpful in getting identities from CCTV cameras that can police to identify the person in no time. It can also be implemented in home security systems as well.

References

[1] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, volume 1, pages I–511. IEEE, 2001.

[2] Paul Viola and Michael J Jones. Robust real-time face detection. International journal of computer vision, 57(2):137–154, 2004.

[3] T. Mita, T. Kaneko, O. Hori, Joint Haar-like Features for Face Detection, “Proceedings of the Tenth IEEE International Conference on Computer Vision”, 1550- 5499/05 ©2005 IEEE.

[4] Jianguo Li and Yimin Zhang. Learning surf cascade for fast and accurate object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3468–3475, 2013.

[5] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li. Aggregate channel features for multi-view face detection. In Biometrics (IJCB), 2014 IEEE International Joint Conference on, pages 1–8. IEEE, 2014.

[6] Shengcai Liao, Anil K Jain, and Stan Z Li. A fast and accurate unconstrained face detector. IEEE transactions on pattern analysis and machine intelligence, **38**(2):211–223, 2016.

[7] Davis E. King. Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research, **10**:1755–1758, 2009.

[8] Michael Jones and Paul Viola. Fast multi-view face detection. Mitsubishi Electric Research Lab TR-20003-96, 3:14, 2003.

[9] Xiangxin Zhu and Deva Ramanan. Face detection, pose estimation, and landmark localization in the wild. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 2879–2886. IEEE, 2012.
[10] XiaohuiShen, Zhe Lin, Jonathan Brandt, and Ying Wu. Detecting and aligning faces by image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3460–3467, 2013
[11] C. Garcia and M. Delakis. A neural architecture for fast and robust face detection. In Pattern Recognition, 2002. Proceedings. 16th International Conference on, 2002. 2
[12] M. Osadchy, Y. L. Cun, M. L. Miller, and P. Perona. Synergistic face detection and pose estimation with energy-based model. In In Advances in Neural Information Processing Systems (NIPS), 2005.