Face Detection

Koishore Roy, Ujjwal Yadav

Ashoka University, Plot #2, Rajiv Gandhi Education City, PO Rai and Teh, Haryana, Sonipat-131029

16 December 2016

ABSTRACT: Face detection is a computer technology that identifies a human face within digital images. This project is a python script which, when run allows the user to take pictures using the Spacebar key and exit at will using the Escape key. Those pictures are then stored in a folder in the root directory of the script, with rectangles drawn in different colours around all the faces that have been detected.

Keywords: face detection, machine learning, haarcascade, frontal face, features, image, webcam.

1. Introduction

1.1. Intro:

Face detection is a computer technology that detects whether a human face is present within a digital image or not. Face detection algorithms primarily focus on detecting a frontal human face. The features are tested and through these tests can the program identify a face.

1.2. Description:

Firstly, the conceivable human eye locales are distinguished by testing all the valley districts in the grayscale picture. At that point the genetic algorithm is utilized to produce all the conceivable face areas which incorporate the eyebrows, the iris, the nostril and the mouth corners.

Every conceivable face applicants is standardised to lessen lightning impact brought about because of uneven brightening and the shirring impact because of head development. The wellness estimation of every competitor is measured in light of its projection on the Eigen - faces. After various emphases, all the face competitors with a high wellness esteem are chosen for further confirmation. At this stage, the face symmetry is measured and the presence of the distinctive facial elements is checked for every face competitor.

1.3. Uses:

Face detection is used in biometrics, often as a part of (or together with) a facial recognition system. It is also used in video surveillance, human computer interface and image database management.Some recent digital cameras use face detection for autofocus. Face detection is also useful for selecting regions of interest in photo slideshows that use a pan-and-scale Ken Burns effect. Modern appliances also use smile detection to take a photograph at an appropriate time.

2. Method

2.1. *Algorithm:*

First, the python script launches a window, which accesses the webcam to take pictures. A basic infinite loop to take images at will, with an 'Escape to quit' protocol has coded into the window. When the Spacebar is pressed, the webcam image is read and written onto a file in the format 'facedetect{n}.png' where n represents the total number of pictures taken so far. This picture is then processed through another python script, which has a function called 'detect'. This function takes a photo and find faces from it and in all the faces found, draws a different randomly coloured rectangle. The image with the detected face then overwrites the current file and is saved within a directory named, "images."

2.2. *Manual input:*

There is also a way to manually input photos into the system and detect faces within the photo. Run the python script named 'face_detectmanual.py', and it will give a similar kind of output. The script, 'face_detectmanual.py' asks for the name of the picture you want to scan for faces. It then asks for the extension of the image. After the user inputs both the required details, it works similar to the 'face_detect.py' script, and scans the given image for human faces using the "haarcascade_frontalface.xml" and outputs a photo with scanned faces and with different coloured rectangles drawn around them.

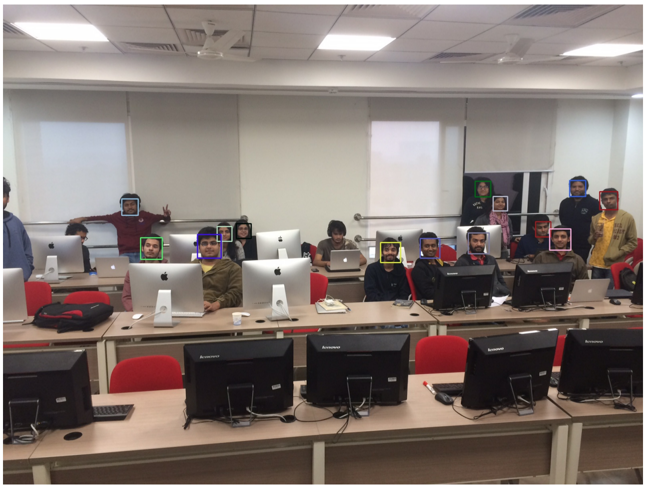

Fig. 2.3a. *Sample image to show the output.*

2.3. *Presentation*

During the Introduction to Machine Learning expo, a picture of the whole class was taken. The picture was then run through our code and it output the faces of the people that were fully visible and had a straight line of sight at the camera. Fig. 2.2a is the output of the picture taken and run through our script.

3. Discussion and Conclusion

3.1. *Other Scopes*

This is a very basic machine learning project and has immense scopes and areas to improve and add upon. The face detection algorithm may be added upon by having a gender recognitionn algorithm in it, where a set of male faces and female faces are trained and then tested against a male or female face to check if the code can predict whether it is male or female. Otherwise, a face recognition algorithm would have been a huge plus to this script, which would test a database of faces (example, the Yale face database) and check with a testing face whether the face is in a given database or not. This may be used for an attendance marking system, a social media tagging system, surveillance, biometric security etc. But due to time constraints, we could implement a script that gives confidence intervals of how close a face is to the testing face in a database. We decided not to include that as it doesn't add anything significant to our project, taking away from the presentation during the expo.

3.2. *Conclusion*

It can be seen that face detection is an essential stepping stone for much larger machine learning projects such as gender detection, facial recognition etc. Even under these categories, there are various other fields that one may use face detection in.

4. Acknowledgement

We would like to thank our professors, Dr. Yamuna Prasad, Dr. Ravi Kothari, and Dr. Raghav Singh for teaching us the basics of machine learning and giving us an intro to the vast field of machine learning.

5. References

Face detection, n.d. *In Wikipedia*. Retrieved December 15, 2016 from *https://en.wikipedia.org/wiki/Face_detection*

Ken Burn effect, n.d. *In Wikipedia*. Retrieved December 15, 2016 from *https://en.wikipedia.org/wiki/Ken_Burns_effect*

The Yale Face Database, n.d. *UCSD Computer Vision*. Retrieved 15 December, 2016 from *http://vision.ucsd.edu/content/yale-face-database*

Other Samples Outputs of our Code