International Journal of Advance Research, Ideas and Innovations in Technology

INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 7, Issue 3 - V7I3-1958)
Available online at: https://www.ijariit.com

Smart Selfie using Computer Vision

Sumit Bajare
sumitbajare333@gmail.com
Dhole Patil College of Engineering, Pune, Maharashtra

Prajval Shende
prajval.shende@gmail.com
Dhole Patil College of Engineering, Pune, Maharashtra

Chaitanya Madgundi
madgundichaitanya1234@gmail.com
Dhole Patil College of Engineering, Pune, Maharashtra

Bhargav Bhatt
bhargavbhatt777@gmail.com
Dhole Patil College of Engineering, Pune, Maharashtra

Rucha Joshi
Assistant Professor
Dhole Patil College of Engineering, Pune, Maharashtra

ABSTRACT

In this research, we will use computer vision to implement real-time face detection and tracking of the lips position from high-definition video. As Face detection and recognition from an image or a video is a popular topic in biometrics research, it has widely attracted attention due to its enormous application value and market potential, such as real-time video surveillance system. For face detection and tracking the location of the lips, we use the OpenCV library. The experimental findings were produced utilising computer vision and OpenCV framework libraries at 30 frames per second at 1080p resolutions for increased accuracy and speed in face detection and tracking of lips position, as well as taking a high-definition smart selfie.

Keywords— *Face detection, Face Recognition, OpenCV*

1. INTRODUCTION

Face detection is a type of computer algorithm that determines the locations and sizes of human faces in digital photographs. Face recognition technology has gotten a lot of press because of its vast range of applications and business possibilities, including real-time video surveillance systems. Law enforcement and military personnel are increasingly using it for forensics. Face recognition is widely acknowledged to have played an essential part in surveillance systems because it does not require the co-operation of the object. Face detection just looks for facial traits and ignores everything else, including buildings, trees, and people. Human face perception is a hot topic in the computer vision world right now. The recognition and localization of human faces is frequently the most difficult task. Human face perception is currently an active research area in the computer vision community. Human face localisation and detection is often the first step in applications such as video surveillance, human computer interface, face recognition and image database management.

Woodrow Wilson Bledsoe pioneered facial recognition in the 1960s. Bledsoe created a technology that could classify photographs of faces by hand using a RAND tablet, a gadget that allowed individuals to input horizontal and vertical coordinates on a grid with the help of a pen-like stylus that emitted electromagnetic pulses. In this paper we intend to implement the Haar-Classifier for Face detection and tracking based on the HaarFeatures. We are going to detect the face which gets into the field of vision of the camera and save the face in the form of an image in JPG format.

2. EXISTING WORK

Face detection in real time is critical in a variety of applications, including biometrics, video surveillance, human computer interfaces, and picture database management. Face detection is used for autofocus in several contemporary digital cameras. Marketers are becoming more interested in face detection. Any face that walks by can be detected by a webcam that is incorporated into a television. Face recognition is also possible. Face detection is also being researched in the area of energy conservation.

Face detection is also employed by the government of the United States of America at airports, where facial detection and recognition technologies can monitor persons arriving and departing. The technology has been utilised by the Department of Homeland Security to identify people who have overstayed their visas or are under criminal investigation. Customs officials at Washington Dulles International Airport made their first arrest using facial recognition in August 2018, catching an impostor trying to enter the country.

Apple was the first to employ facial recognition to unlock the iPhone X, and it continues to do so with the iPhone XS. Face ID verifies that you are who you say you are when you access your phone. According to Apple, the chances of a random face unlocking your phone are one in one million.

Face detection is used in Businesses at entrances and restricted areas. Some companies have traded in security badges for facial recognition systems. Beyond security, it could be one way to get some face time with the boss.

3. DESCRIPTION OF TOOLS

In this section, the tools and methodology to implement and evaluate face detection and tracking using OpenCV are detailed.

3.1 OPENCV

OPENCV (Open-Source Computer Vision Library) is a computer vision and machine learning software library that is free to use. OpenCV was created to provide a common infrastructure for computer vision applications and to help commercial goods incorporate machine perception more quickly. Because OpenCV is a BSD-licensed product, it is simple for businesses to use and alter the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognise faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken with flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

OpenCV has more than 47 thousand people of user community and estimated number of downloads exceeding more than 18 million. The library is used extensively in companies, research groups and by governmental bodies.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. OpenCV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available. A full-featured CUDA and OpenCL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. OpenCV is written natively in C++ and has a templated interface that works seamlessly with STL containers.

3.2 ADABOOST

A concept called Adaboost which both selects the best features and trains the classifiers is used. This algorithm constructs a strong classifier using a linear combination of weighted simple weak classifiers. The poor learners (rules of thumb) must have an error rate of less than 50%, which means they must be marginally better than random guessing. Simply invert the sign of the rule of thumb if it has more than 50% error. It's easier to identify and combine numerous simple (weak) rules of thumb than it is to find one complex (correct) one. Freund and Schapire designed Adaboost in 1997.They won the Gödel prize for this contribution in 2003. Adaboost was applied to face detection (with some modifications) by Viola and Jones in 2001.

3.3 HAAR Feature Selection

Paul Viola and Michael Jones proposed Haar Cascade as a machine learning object detection approach in their paper "Rapid Object Detection with a Boosted Cascade of Simple Features" in 2001. It's a machine learning strategy in which a cascade function (which I'll discuss later) is taught from a large number of positive and negative photos (where positive images are those where the object to be detected is present, negative are those where it is not). After then, it's utilised to find items in other photos. Fortunately, OpenCV provides pre-trained Haar cascade algorithms that are sorted into categories (faces, eyes, etc.) based on the images they were trained on. The first step is to get the Haar Characteristics. A Haar feature evaluates neighbouring rectangular sections in a detection window at a specified position, adds the pixel intensities in each region, and computes the difference between these sums.

4. REQUIREMENT

4.1 Software Requirement

Python (3.8 version or later).
GUI: Anaconda Navigator. (2.0 version or later)
Python libraries like OpenCV (3.4.0 version or later), Matplotlib (3.3.1 version or later) , Numpy, dlib (19.21.1 version or later), scipy (1.6.1 version or later), imutils (0.5.4 version or later), time Libraries.

4.2 Hardware Requirement

PC preferably running windows with Camera with video quality of 320p to capture a clear selfie.
Processor should be intel i5 or higher as to execute a computer vision algorithm a greater computation power is required.
500GB SSD, 4GB RAM Memory sends information to other components at a faster rate with more RAM. As a result, a fast CPU now has an equally rapid way of communicating with the other components, resulting in a considerably more efficient computer.

5. PROPOSED SYSTEM AND TECHNIQUE

The accuracy of the system determines the quality of the selfie. To standardise the photos, you submit to a face recognition system, you should use a variety of pre-processing techniques. Under great light sensitivity, face recognition algorithms have a hard time recognising a face. If the system has been programmed to recognise a human in a dark room, then it is highly possible that it won't recognise them in a bright room. This problem is referred to as “illumination dependent". There are many other issues, such as the face should also be in a very consistent position within the images like the eyes being in the same pixel coordinates, consistent size, rotation angle, hair and makeup, emotion like smiling, angry, etc. As a result, selecting a decent image pre-processing filter is critical. To make things easier, convert colour photos to grayscale and then use Histogram Equalization. It's a pretty simple way to standardise the brightness and contrast of your facial photos automatically. Apply extra processing steps, such as edge enhancement, contour detection, motion detection, and so on, for better results. OpenCV uses a face detector algorithm called a Haar Cascade classifier. The image comes from live video, the face detector examines the face and the position the lips. Once the smile is detected the system will capture the photo in JPEG format. Hence a selfie is clicked.

6. SAMPLE CODE

6.1 For determining the lips position

```
def smile(mouth):
    A = dist.euclidean(mouth[3], mouth[9])
    B = dist.euclidean(mouth[2], mouth[10])
    C = dist.euclidean(mouth[4], mouth[8])
    avg = (A+B+C)/3
    D = dist.euclidean(mouth[0], mouth[6])
    mar=avg/D
    return mar
```

6.2 For Face detection

```
while True:
    frame = vs.read()
    frame = imutils.resize(frame, width=450)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rects = detector(gray, 0)
    for rect in rects:
        shape = predictor(gray, rect)
        shape = face_utils.shape_to_np(shape)
        mouth= shape[mStart:mEnd]
        mar= smile(mouth)
        mouthHull = cv2.convexHull(mouth)
        #print(shape)
        cv2.drawContours(frame, [mouthHull], -1, (0, 255, 0), 1)
```

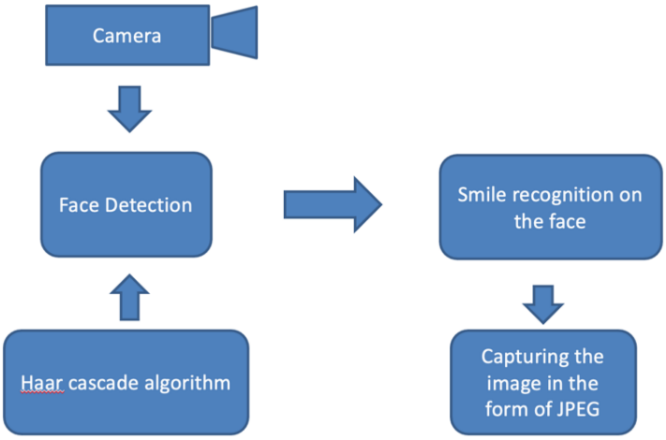
6.3 For Auto Selfie Capture:

```
if mar <= .3 or mar > .38 :
        COUNTER += 1
    else:
        if COUNTER >= 15:
            TOTAL += 1
            frame = vs.read()
            time.sleep(.3)
            frame2= frame.copy()
            img_name = "opencv_frame_{}.png".format(TO-
TAL)
            cv2.imwrite(img_name, frame)
            print("{} written!".format(img_name))
        COUNTER = 0
```

6.4 Libraries used

(a) OpenCV
(b) Dlib
(c) Imutils
(d) Time
(e) Scipy
(f) Numpy
(g) Matplotlib

7. SYSTEM ARCHITECTURE



8. CONCLUSIONS

Computer Vision, a subset of Deep learning is the field that have been researched a lot, so in the coming future, more advance features will be added to the system. With advancements in technology, real-time face detection in remote monitoring is becoming more beneficial in the development of many efficient industrial and commercial applications. Moreover, this technology is useful in application such as uploading a phone which provides unique customer experience. This study could be improved by using stereo depth analysis of face detection with two image sensors connected to a high-speed processor.

9. REFERENCES

[1] https://opencv.org/about/
[2] https://www.cse.iitb.ac.in/~ajitvr/CS763_Spring2017/Adaboost_FaceDetection.pdf
[3] https://medium.com/dataseries/face-recognition-with-opencv-haar-cascade-a289b6ff042a
[4] https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
[5] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html
[6] https://matplotlib.org
[7] http://dlib.net
[8] https://www.scipy.org
[9] https://en.wikipedia.org/wiki/AdaBoost
[10] https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/

