Smart Selfie Based on Computer Vision

SANGAMESH HOUSHETTY[1], SABA SHIREEN[2], SHABARI RATHOD[3], SHABISTA MEHRISH[4]
[1, 2, 3] *Department of CSE, GNDEC, Bidar*

Abstract- In this Project, 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 sur-veil lance system. For face detection and tracking the location of the lips, we use the OpenCV library. The experimental findings were produced utilizing computer vision and OpenCV framework libraries at 30 frames per second at 1080p resolutions for increased accuracy and speed. lance system. For face detection and tracking the location of the lips, we use the OpenCV library. The experimental findings were produced utilizing 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.

I. 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 localization 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 photo-graphs of faces by hand using a RAND tablet, a gadget that al-lowed 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 in

II. LITERATURE SURVEY

Development of research related to computer vision and digital image processing

- Researchers: -Savioja et al., [15], Fadeyev, V., & Haber, C. [16]
- Usefulness: -image processing applied in the digital sound system
- Definition: -Audio information stored in the flowing threads in the media such as phonograph recording can be reconstructed, with no minimal contact, by measuring the shape of the groove using precision metrology and digital image processing methods.
- Researchers: -Various, N. A., & Sun, D. W. [17] (2012) Langton et al. [18], Li et al. [20], Jackman et al. [19], Du and Sun [21], Jusoh et al. [22], Pallottino et al. [23]
- Usefulness: -image processing used in food analysis
- Definition: -It has three main steps, e.g., feature selection and extraction, and classification. Feature selection and retrieval, and classification. This approach will be reviewed from aspects of role portrayal and its impact on the food and beverage industry.
- Researchers: -Canny [24]; Vacchetti et al., [25] Wang [26], Zheng et al., [27]
- Usefulness: -Convolutional Neural Networks (CNN) is used for object detection.

- Definition: -Use of ANN in in-depth learning to solve edge detection problems in image processing areas

III. EXISTING WORK

Face detection in real time is critical in a variety of applications, including biometrics, video surveillance, human computer inter-faces, 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 utilized by the Department of Home-land 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 un-locking 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

IV. PROPOSED SYSTEM

The accuracy of the system determines the quality of the selfie. To standardize the photos, you submit to a face recognition system, you should use a variety of pre-processing techniques. Un-der great light sensitivity, face recognition algorithms have a hard time recognizing a face. If the system has been programmed to recognize a human in a dark room, then it is highly possible that it won't recognize 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 co-ordinates, 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 standardize the brightness and contrast of your facial photos automatically. Apply extra processing steps, such as edge enhancement, contour detection, motion de-taction, and so on, for better results. OpenCV uses a face detect-tor 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.

V. METHODOLOGIES

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

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. 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.

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. Adaboost was applied to face detection (with some modifications) by Viola and Jones in 2001.

HAAR Feature Selection • Paul Viola and Michael Jones proposed Haar Cascade as a ma-chine 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 is those where it is not). After then, it's utilized to find items in other photos.

Machine learning • Machine learning (ML) is a type of artificial intelligence (AI) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning algorithms use historical data as input to predict new output values. There are three machine learning types: supervised, unsupervised, and reinforcement learning. Supervised machine learning models are trained with labelled data sets, which allow the models to learn and grow more accurate over time. Python • Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems. • How Python is used in machine learning? o The Python library provides base-level items, so developers do not have to write code from scratch every time. Machine learning requires continuous data processing, and Python libraries allow you to access, process, and transform your data.

VI. PROPOSED METHODOLOGY

Facial Feature Extraction There are multiple models which are used for detection of face and extraction of facial features from it. One such example of facial landmark detection [13] model present in open source is Dlib's68-facelandmark: "shape_predictor_68_face_landmarks.dat"[14]. Dlib's 68-face landmark model shows how we can access the face features like eyes, eyebrows, nose, etc. But sometimes, we don't want to access all features of the face and want only some features likes, lips for lipstick application. So that is also possible using custom training of the Dlib's 68- landmark models. There are multiple other models with different number of facial plot points and as we increase or decrease the number of plots points the accuracy also increases or decreases. Some of the models used by us in this research paper

Face Points vector Computation The next step after facial points extraction was to normalize the facial feature points, so as the scale for both template image points as well as the bitmap captured by camera sensor were on same scale and comparable for next process of similarity computation. We used point 17 of the 35-point model to compute the vectors for all points. Vxi = Xi – X17 and VYi = Yi – Y17 e.g. , Vx0 = XX0 – X17 and VY0 = Y0 – Y17 Vx1 = X1 – X17 and VY1 = Y1 – Y17 Vx2 = X2 – X17 and VY2 = Y2 – Y17 Vx3 = X3 – X17 and VY3 = Y3 – Y17 . .. Vx33 = X33 – X17 and VY33 = Y33 – Y17 Vx34 = X34 – X17 and VY34 = Y34 – Y17 Where , Vxi = x-component of vector V for point i, Vyi = y-component of vector V for point i, Xi = x-coordinate for point i, Yi = y-coordinate for point i. Vector Vi (magnitude) = sqrt (Vxi2 + Vyi2) Vector Vi (direction) = VYi / Vxi ➢ Vector Selection Vector selection is an important procedure before computation of similarity between template and sensor data[15]. In case if all vectors are considered it will increase the similarity score between the faces. It will happen as there are many points in face which are same for different facial expressions i.e., facial points corresponding to fixed facial parts and do not contribute in making different expression using our face, e.g. – Jawline points (point 26 to 34 for 35-point model) (not all points were excluded 29, 30 and 31 were included for similarity). There are the points which only vary in different person based on face shape and therefore may be different for different people and hence can also reduce the similarity expression for similar expressions.

Cosine Similarity Cosine similarity measures the similarity between two vectors [16] of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis. A document can be represented by thousands

of attributes, each recording the frequency of a particular word (such as a keyword) or phrase in the document. Thus, each document is an object represented by what is called a term frequency vector. Cosine similarity is a measure of similarity that can be used to compare documents or, say, give a ranking of documents with respect to a given vector of query words. Let x and y be two vectors for comparison. Using the cosine measure as a similarity function, we have Sim(x, y) = x.y / ||x|| * ||y|| where, ||x|| is the Euclidean normal of vector x=(x0, x1 ,…, xn), defined as sqrt(x02 + x12 + x22 + …. + xn2). Conceptually, it is the length of the vector. Similarly, ||y|| is the Euclidean norm of vector y. The measure computes the cosine of the angle between vectors x and y. A cosine value of 0 means that the two vectors are at 90 degrees to each other (orthogonal) and have no match. The closer the cosine value to 1, the smaller the angle and the greater the match between vectors. Similarity = cos (θ) = (A.B) / ||A|| ||B|| = (Σ Ai * Bi) / sqrt Σ (Ai) 2 * sqrt Σ (Bi) 2 Fig 5: Representation of formula for cosine similarity When attributes are binary-valued, the cosine similarity function can be interpreted in terms of shared features or attributes. Suppose an object x possesses the i the attribute if xi = 1. Then x t · y is the number of attributes possessed (i.e., Sim (x, y) = 0.94 ➢ Face Orientation Error Face orientation[17] similarity is the check for comparing the similarity in orientation of both the template face and bitmap from sensor i.e., the direction in which both the faces are looking or in which direction are the faces tilted. There are two types are orientation which are computed. These are:-

- Left-Right: The left-right orientation is computed by calculating ratio of distance of x-coordinates between points 7 and 14 to 14 and 11 ((x7 – x14) / (x14-x11)). Ideally this value should be close to 1 when a person is looking straight into the camera. When the person's head tilts towards left the value of left-right orientation ratio becomes greater than 1 and keeps on increasing as the person keeps tilting its head towards left. Similarly, when a person looks towards right the value of left-right orientation ratio becomes lesser than 1 and keeps on decreasing as the person keeps tilting its head towards right.
- Up-Down: Similar to the left-right orientation ratio, in up-down orientation ratio of distance of coordinates between points 14 and 17 to 17 and 30 ((y14 – y17) / (y17-y30)). Ideally this value should be close to 0.5 when a person is looking straight into the camera. When the person's looks up the value of up-down orientation ratio becomes less than 0.5 and keeps on decreasing as the person keeps tilting its head upwards. Similarly, when a person looks down the value of up-down orientation ratio becomes greater than 0.5 and keeps on increasing as the person keeps tilting its head downwards. The error for both the ratios are calculated by:- Elr = (LRsensor - Template) / LRtemplate Eud = (UDsensor - UDemplate) / LRtemplate Here, Elr = Error left-right orientation ratio LRtemplate = Left-right orientation ratio for template face LRsensor = Left-right orientation ratio for sensor face Eud = Error up-down orientation ratio UDemplate = Up-down orientation ratio for template face UDsensor = Up-down orientation ratio for sensor face The root mean squares of both the errors are compared using formula:- RMSE = sqrt(Elr * Elr + 4 * Eud * Eud) Here a factor of 4 is multiplied to Eud its value is normal case is close to 0.5 whereas value of Elr in normal case is close to 1 and we are squaring both the errors before taking square root.

Computing result by similarity and error score and capturing image Once we have got both the cosine similarity score and face orientation error score we compare it to the cosine similarity score and face orientation error score of previous captured image[18]. When no image has been saved, the initial value is:- Cosine similarity score = 0 Face orientation error score = Integer. MAX_VALUE. If both the score of bitmaps is better than previous captured image we send that bitmap for processing and save that image into the memory and replace cosine similarity score and face orientation error score with the values of current bitmap which is saved.

VII. REQUIREMENTS

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)

- Dlib (19.21.1 version or later),
- Scipy (1.6.1 version or later),

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.

CONCLUSION

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 ap-plications. 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.

ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our Prof. GURUPRASAD as well as our HOD Prof. DAYANAND.J who gave us the golden opportunity to do this wonderful project on the topic "SMART SELFIE BASED ON COMPUTER VISION" which also helped us in doing a lot of Research and we came to know about so many new things we are really thankful to them. We are also thankful to all the other faculty, teaching and non-teaching staff members of our department for their kind co-operation and help.

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_classi-fier.html

[5] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detec-tion/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-ada-boost-for-machine-learning/