International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 06 | June - 2022 Impact Factor: 7.185 ISSN: 2582-3930

Facial Recognition system with voice message enhancement

Jonnadula Narasimharao[1], CH. Lakshmi Sravya[2], S. Tharun[3], Parasa Rithvik[4]

[1]Associate Professor of Dept of CSE, CMR Technical Campus, Hyderabad, India

[2]UG Student, CMR Technical Campus, Dept of CSE, Hyderabad, India

[3]UG Student, CMR Technical Campus, Dept of CSE, Hyderabad, India

[4]UG Student, CMR Technical Campus, Dept of CSE, Hyderabad, India

---***---

Abstract: Face recognition system has always been a crucial human-computer interaction tool in most security systems, access control, video surveillance, commercial areas, and networks like Facebook as well. After development in the area of artificial intelligence and growing security concerns, face recognition system has attracted attention by providing maximum efficiency in face detection and recognition. Face recognition is considered to be a biometric artificial intelligence-based application. Face recognition is the process of recognizing the face of a person by a vision system. As an extension to the existing face recognition system in this model, a new feature is added i.e., a voice message as output that is produced for each face recognized. The voice message could be anything like a simple hello or even a complex sentence.

Keywords: Artificial intelligence, Face recognition, voice message, OpenCV, pyttsx.

1. INTRODUCTION

Face recognition has always been a fast-growing, challenging, and interesting area in real-time applications in biometric security applications. The paper presents an extension to the existing facial recognition system with a voice message for every recognized face. Today a lot of applications in various sectors use face recognition for security purposes like surveillance systems and thus this stands out as an application with maximum utilization as part of artificial intelligence. A large number of face recognition algorithms have been developed in the last few years which continue to serve many purposes. Traditional methods have been facing challenges like pose variation, facial features, lighting of the scene, the complexity of the image background, changes in facial expression, etc. Deep learning-based methods can extract more complicated face features and there are different deep learning approaches like Convolutional Neural Network (CNN), principal component analysis, and support vector machine. CNN has mostly used algorithms in image and face recognition. In the developed model, we are concentrating on using computer vision (OpenCV) as the basis for face recognition in our project and OpenCV for face detection and face recognition using the video frame sequence. Classifiers like haar cascade have been used for the face detection process in this project.

The Face_recognition package from python has been used for the face recognition part. Modules like pyttsx for the conversion of text to speech and a voice message as output for every face recognized are given. The model has an accuracy of over 99.38% that provides a simple command-line tool that helps us perform face recognition on a database of images. Using OpenCV is a much easier yet effective tool for face detection and recognition.

2. LITERATURE SURVEY

Face recognition system development started in the early 1950s and 60s. Initially face recognition system was a semi-automated system where the distance between landmarks was automatically computed to determine identity. A lot of algorithms since then have been developed to implement how to detect faces and recognize them. One of the most popular methodologies includes how face recognition can be implemented using eigenfaces [7] which describes the extraction of information in images independent of any features and using this information to encode and decode the face images. Since then, various face recognition algorithms, concentrated on much better image quality and have become active applications of pattern recognition, and image analysis. Another algorithm named PCA [1-7] also uses the eigenfaces technique for the implementation of face detection and recognition. Techniques like LDA [1-3] have also resulted to be dominant in feature selection and dimension reduction. SVM (support vector machine) [1] methodology was also able to achieve high generalization performance which is considered to be one of the most effective algorithms for pattern classification problems. OpenCV [2] has been a powerful tool giving accuracy at a larger scale compared to the traditional face recognition systems.

Face recognition not only has been developed by the above algorithms but can be implemented with the usage of

Raspberry Pi [8] which displays the matched status of the images on the LCD of the raspberry pi. The ai approach uses artificial neural networks [9] for face recognition. In this multi-layer perceptron was used to classify the results. Some researchers also have solved the problem of multi-class face recognition problems into a number of binary classification problems [9]. A lot of other approaches over time also have been used for the implementation of face recognition and some of them include the LBP (local binary pattern) histograms [9] where the facial features are selected by the LBP histograms of the regions of the images of faces the distance corresponding these histograms are chosen to be the features and an algorithm called Ada boost was applied to select the most efficient LBP features out of it. Hidden Markov models [9] have also been utilized for the face recognition task. And using the hidden Markov model accuracy of over 87% was seen.

3. PROPOSED METHODOLOGY

We propose a system that recognizes people from a video frame sequence. More specifically we are interested in locating the images of people and producing a voice message each time the face of the person is recognized accurately. A database of images is already fed to the system as input with the images of all the people whom we wish to implement this model with, to match against the faces from the live video capture. With the help of python libraries like face recognition and computer vision, the recognition part is implemented. A live video capture box opens as soon as the application is opened and starts to run, which in turn automatically is responsible for the camera to be on and is ready to capture the person/people in the frame and produce the desired output with a voice message [6]. This voice message could be anything like a simple hello or a complex statement depicting a person/people's personal information. This model is implemented and tested at an institution-level where images have been classified into three major categories namely management, professors, and students. The output for each particular module would output a different message each time it has recognized a face. For example, if the face of a student in the institution is recognized a message such as, “good morning student, your room number is 201” would be given as an output. In case of a mismatch, a common message to all the modules would be outputted to display that the face does not exist in the database.

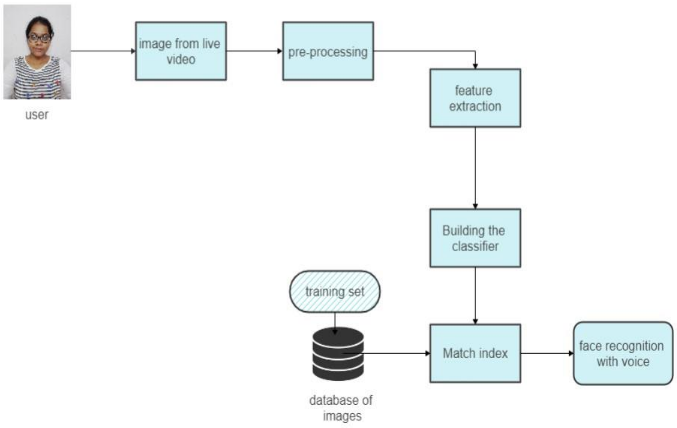


Fig-1: Overview of the facial recognition system process

3.1. Face recognition from python

Built using dlib’s state of art recognition, it provides a simple Face_recognition library that helps us perform face recognition on a database of images. This Face_recognition lets us perform the face recognition from the command line too. It has an accuracy of over 99.38% and the database or the training model that needs to be given as an input just has one image of every person that needs to be recognized.

1. **Face detection**
   The first task is to detect the faces, and this is done from a video stream whenever the application is run to start. The images are detected for further processing for the face recognition part. A rectangular box opens up around each face that is detected in the video frame sequence. Continuous detection of faces is carried out until the application is closed. You can automatically find all the faces in the image.

2. **Finding out the facial features**
   We can automatically locate the facial features in this step which is the most crucial process in face recognition. As facial features are unique and with this step, we can get the locations and outlines of every person’s eye, nose, mouth, and chin area. This is called the marking of facial landmarks.

3. **Comparing and identification of faces.**
   Now we identify who appears in each photo and with the help of the known facial landmarks from the above step and the images with which the model is already trained (training dataset) the faces are compared and a match or a mismatch is declared.

4. **Voice message as output.**
After the comparison of the known and unknown faces and a match is made, the output is a voice message that is produced. And the voice message is a predefined message that the database is already trained with for every image that is recognized. Using the library pyttsx3 this is implemented.

3.2. Using the pyttsx module for voice generation

Pyttsx3 is the python module that is responsible for producing the text-to-speech conversion. It is one of the easiest tools for the conversion of text to speech effectively. This module can support both female and male voices, but the male voice is possible with the help of sapi5 for windows. Using the import pyttsx3 the module can be installed onto the local device and be used. The output would be a voice saying the text that is given to it as input. In this project, we have used the pyttsx3 module for the text that is present in the database to be converted to a voice message for every face recognized.

4. RESULTS

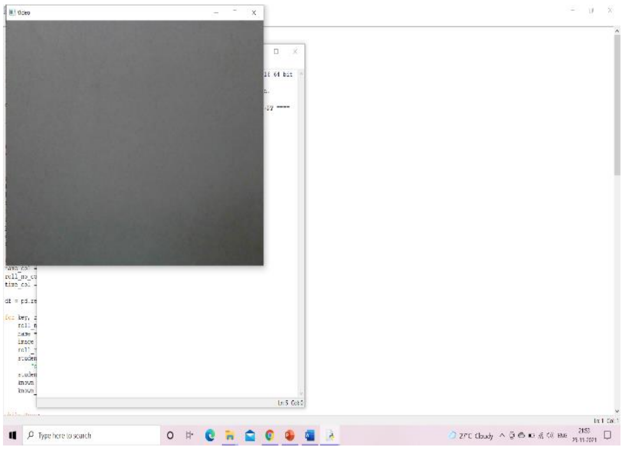
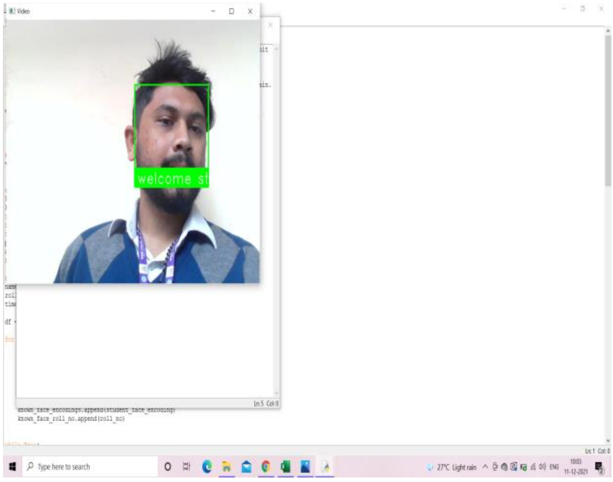


Fig-2: The live video capture frame
The video frame is the start point of the application and opens as soon as the code is run from the prompt.



```
known_face_encodings.append(student_face_encoding)
known_face_roll_no.append(roll_no)
```

Fig-3: Image of the face recognized
The person is recognized and a message “welcome student” is given as a voice output that is displayed on the console.

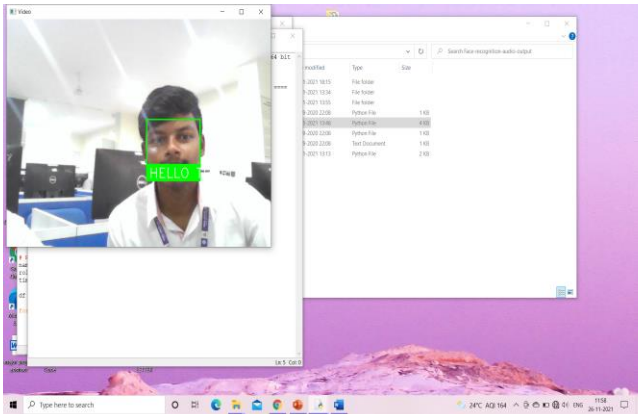
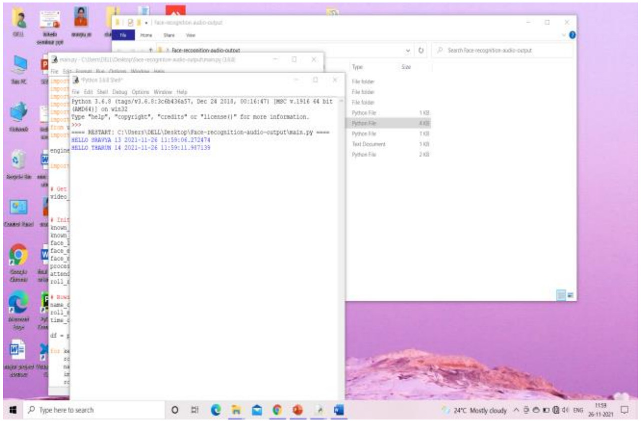


Fig-4: Results of the recognized face



```
Python 3.6.8 (tags/v3.6.8:3c6b436a57, Dec 24 2018, 00:16:47) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: C:\Users\DELL\Desktop\Face-recognition-audio-output\main.py ====
HELLO SRAVYA 13 2021-11-26 11:59:06.272474
HELLO TARUN 14 2021-11-26 11:59:11.987139
```

Fig-5: The log file created for various inputs

6. CONCLUSION

The paper reflects an innovative approach to adding features to the face recognition system with a voice message as output. The model can effectively deliver voice messages to various faces which are trained with images in the database to give a much more efficient system with just one image being trained. This could be a potential addition to the security domain of various applications using biometric security within them. The model can successfully deliver information that it is trained to output for a particular image with 99% accuracy.

ACKNOWLEDGMENT

The success of this paper includes help from our guide as well. We are grateful to our guide, Mr. J. Narasimharao, Associate professor, CMR Technical Campus, for his expertise that assisted us in our research.

REFERENCES

[1] S. G. Beale and V. H. Mankar, "A Review Paper on Face Recognition Techniques," Int. J. Adv. Res. Comput. Eng. Technol., vol. 1, no. 8, pp. 2278–1323, 2012.

[2] M. Naveenkumar and A. Vadivel, "OpenCV for Computer Vision Applications", Proceedings of National Conference on Big Data and Cloud Computing (NCBDC'15), March 20, 2015.

[3] D. N. Parmar and B. B. Mehta, "Face Recognition Methods & Applications," Int. J. Comput. Technol. Appl., vol. 4, no. 1, pp. 84–86, 2013.

[4] Raktim Ranjan Nath, Kaberi Kakoty, and Dibya Jyoti Bora, "Face Detection and Recognition Using Machine Learning Algorithm".

[5] Nurulhuda ismail, Mas idayu md. Sabri, "REVIEW OF EXISTING ALGORITHMS FOR FACE DETECTION AND RECOGNITION". ISSN: 1790-5117

[6] https://pyttsx3.readthedocs.io/en/latest/engine.html

[7] Mathew A. Turk and Alex P. Pentland, "Face recognition using eigenfaces", Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91., IEEE Computer Society Confer

[8] Ishita Gupta, Varsha Patil, Chaitali Kadam, and Shreya Dumbre, "Face detection and recognition using Raspberry Pi". In 2016 IEEE International WIE Conference 185 on Electrical and computer engineering (WIECON-ECE), pages 83–86.

[9] Rabia Jafri, and Hamid R. Arabnia, "A Survey of Face Recognition Techniques", Journal of Information Processing Systems, Vol.5, No.2, June 2009 41.