Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic

Jayanth Vadlapati
Amity University
Dubai
jayanthV@amitydubai.ae

Senthil Velan S
Amity University
Dubai
svelan@amityuniversity.ae

Ewin Varghese
Amity University
Dubai
ewinV@amitydubai.ae

2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 978-1-7281-8595-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCNT51525.2021.9579712

***Abstract*— - In this research paper, we are going to see the profound scientific use of computer technology applied in the fields of AI and Machine Learning primarily focused on Image Processing and Pattern recognition. Techniques such as ours are widely used to recognize real life objects including human faces etc. Thus, using such techniques, we can recognize a person from pictures. Using face recognition modules from python's huge collection of libraries, we are able to train the model to recognize people while wearing masks. Since when masks are worn, half of the facial features are lost, therefore developing a technique to recognize faces in such way is crucial. This specific technology of face detection is used in biometrics, video surveillance, etc. Therefore it's at utmost importance to increase the security as well as efficiency whilst making the recognition faster.**

Keywords—Facial recognition, Face Recognition while wearing Masks, OpenCV, Python Image Library.

I. INTRODUCTION

In the past few decade's recognition and identification of people to accurately authenticate has increasingly gained importance in a variety of places like, airports and other government-based locations. They use various techniques from Fingerprinting and Iris scanning systems, to voice and face recognition systems to authenticate people. So is the growing importance to have an accurate prediction in every extreme scenarios.

Focusing on Facial Recognition, it is a way of identifying an individual's identity using their face. This recognition or detection is done using photos, clips, or real time videos of a person. As we all know due to the pandemic, it is mandatory to where masks. Due to these masks facial recognition of a person cannot be obtained accurately.

The main objective of this research paper is to give light over the specific methods and techniques used to process an image and recognize subjects. What differs is, this recognition or detection is done on subjects with and without masks. This could be possible with the help of Python. In Python, there are several libraries that are available for image processing such as OpenCV, Scikit-image, PIL, Mahotas. For our purposes, we have used PIL or Python Image Library.

In modern day, such techniques aid in various tasks either in private or public grounds. It is widely used (in public screenings) to detect people who are being captured in public video surveillance.

The organization of the paper is as follows. Section II states a brief scope of the problem identified and analyzed in this paper. Section III provides a brief information about the similar attempts and related work done by other researchers. Section IV provides information about the methodology adopted in research work done and presented in this paper. Section V is the core part of the paper where the working of our project is explained. It provides the complete information and methods adopted in this project. Section VI provides the details about testing and outcomes or result achieved. Section VII emphasis on the challenges and problems encountered while working on the project. Section VIII contains details about the future scope of our project. Section IX concludes the core findings and analysis of project and Finally, Section X includes the references used for this research work.

II. SCOPE OF THE PROBLEM

Usually when face is to be detected, the algorithm needs a complete structure of the face which is now covered by masks. Many a times, the AI has hard time detecting faces let alone recognizing people. It is quite hard to train the model to detect a face without using the common facial features such as nose, eyes, mouth, chin etc. So, models are trained to look for a face using some other features as a reference. For example, in CCTV the algorithm tries to look for a face by firstly detecting a body. Similarly, for recognizing face of a person firstly the model needs to be trained to learn features of the face without the mask to have some reference to recognize.

III. EXISTING WORK

As widely as Face detection and recognition is used in modern day, it is seen to be very vital in normal day to day life concerning mainly in grounds of security and safety of people. Though this has been used since decades, the accuracy, and the speed at which the results were found were not the same.

There are different methods proposed for facial recognition. Each have their own advantages and disadvantages. Some of these methods are:

(1) Classical Face Recognition Algorithms
(2) Artificial Neural Networks
(3) Face Description Based Methods, etc.

Facial recognition and detection play a vital role in various fields. As there can be more such disease like the COVID-19 in future, it is important to recognize and understand people even if they are wearing masks. To be par with the existing technology and methods, we focus on recognizing human faces covered with facemasks, which is a very common thing practiced during this pandemic.

IV. METHODOLOGY

Any analysis-based technique requires the identification, definition and design of suitable steps followed as a methodology. Fig 1 shows the sequence of steps which are required for facial recognition. Our program basically follows a simple algorithm.

To study the details of a given image different sets of images from different angles, with/without body etc. are provided for reference. For our program we have currently provided 10-15 images for reference per person collected from different sources. These reference images are images without masks.

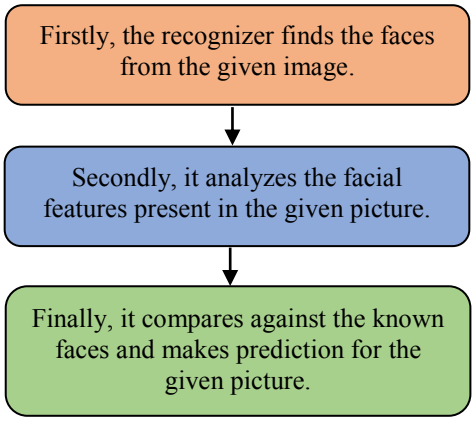

Fig 1. Process Flow of Face Recognition & Detection

Python Programming Language is a free, object-oriented, high-level dynamic language with dynamic semantics. It has high-level built-in data structures combined with dynamic typing and dynamic binding which makes it attractive for Rapid Application Development as well as for scripting or connect existing components together. Python has a huge collection of libraries used for various purposes.

For processes such as Image Learning and recognizing, we have used some of the most prominent and powerful libraries to aid in our process. *face_recognition* is a module used in our program, which is built using Dlib's facial recognition with deep learning and has remarkably high accuracy. Dlib is a contemporary toolkit developed using C++ which contains numerous machine learning algorithms and tools for creating complex software for real world problems. PIL or pillow is another main module used by us, which stands for Python Image Library is immensely powerful library which is used in our program which supports various formats for images like JPG, PPM, PNG etc.

V. WORKING

As said earlier Python has a huge collection of libraries. In this project we have mainly used the python library *face_recognition*. *face_recognition* module is widely used in python for understanding and detecting faces from a given image. It works using the following basic steps,

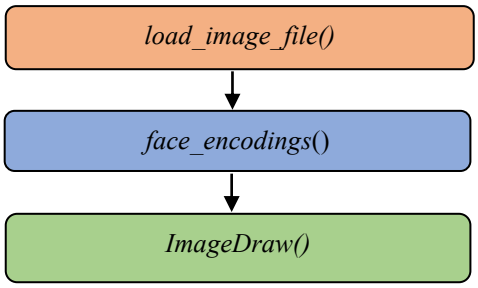

Fig 2. Basic flow of functions

Elaborately, the required images are imported using *load_image_file ()* method, which is converted into NumPy array by the *face_recognition* library. When the face is found, our program tries to find the features found in that specific image, these features are found using an algorithm called face landmark estimation.

```
array([-0.10213576,  0.05088161, -0.03425048, -0.09622347, -0.12966095,
        0.04867411, -0.00511892, -0.03418527,  0.2254715 , -0.07892745,
        0.21497472, -0.0245543 , -0.2127848 , -0.08542262, -0.00298059,
        0.13224372, -0.21870363, -0.09271716, -0.03727289, -0.1250658 ,
        0.09436664,  0.03037129, -0.02634972,  0.02594662, -0.1627259 ,
       -0.29416466, -0.12254384, -0.15237436,  0.14907973, -0.09940194,
        0.02000656,  0.04662619, -0.1266906 , -0.11484023,  0.04613583,
        0.1228286 , -0.03202137, -0.0715076 ,  0.18478717, -0.01387333,
       -0.11409076,  0.07516225,  0.08549548,  0.31538364,  0.1297821 ,
        0.04055009,  0.0346106 , -0.04874525,  0.17533901, -0.22634712,
        0.14879328,  0.09331974,  0.17943285,  0.02707857,  0.22914577,
       -0.20668915,  0.03964197,  0.17524502, -0.20210043,  0.07155308,
        0.04467429,  0.02973968,  0.00257265, -0.00049853,  0.18866715,
        0.08767469, -0.06483966, -0.13107982,  0.21610288, -0.04506358,
       -0.02243116,  0.05963502, -0.14988004, -0.11296406, -0.30011353,
        0.07316103,  0.38660526,  0.07268623, -0.14636359,  0.08436179,
        0.01005938, -0.00661338,  0.09306039,  0.03271955, -0.11528577,
       -0.0524189 , -0.11697718,  0.07356471,  0.10350288, -0.03610475,
        0.00390615,  0.17884226,  0.04291092, -0.02914601,  0.06112404,
        0.05315027, -0.14561613, -0.01887275, -0.13125736, -0.0362937 ,
        0.16490118, -0.09027836, -0.00981111,  0.1363602 , -0.23134531,
        0.0788044 , -0.00604869, -0.05569676, -0.07010217, -0.0408107 ,
       -0.10358225,  0.08519378,  0.16833456, -0.30366772,  0.17561394,
        0.14421709, -0.05016343,  0.13464174,  0.0646335 , -0.0262765 ,
        0.02722404, -0.06028951, -0.19448066, -0.07304715,  0.0204969 ,
       -0.03045784, -0.02818791,  0.06679841])
```

Fig 3. Each of these values represent an orthogonal component of the face encoding. [9]

There are multiple ways to use this algorithm, our program calculates and stores 128 specific points on the face ranging from the top of the chin to the sides of the eyes also covering the eyebrows. This face encoding vector of 128 values is stored in a huge array. The algorithm understands these values as RGB values or certain unique measurements of the face. After training the model by using numerous pictures of the same face at different angles, it becomes thorough in recognizing that specific person.

Now when an image is imported to identify, it will check if the *known_face_encodings* match with any *face_encodings*, where the program cross checks the two arrays and each of the 128 components, later compares the faces and it gives an accurate result if it falls within the tolerance limits and these limits are different for different recognized samples.

```
[array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True, False,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True, False,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True, False,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True, False,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True, False,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True, False,
         True,  True]),
 array([ True,  True,  True,  True,  True,  True, False, False, False,
         True,  True,  True, False,  True,  True,  True, False,  True,
        False,  True,  True,  True,  True, False,  True,  True,  True,
        False,  True,  True,  True, False,  True,  True,  True,  True,
         True,  True,  True,  True, False,  True, False,  True,  True,
         True,  True,  True, False,  True,  True,  True,  True,  True,
        False, False,  True,  True,  True,  True,  True, False,  True,
        False, False, False, False,  True, False,  True, False,  True,
        False,  True,  True,  True,  True, False,  True,  True,  True,
         True,  True,  True, False,  True,  True,  True, False,  True,
         True, False,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True, False, False,  True,  True,
        False, False, False,  True,  True, False,  True,  True,  True,
         True,  True,  True,  True,  True,  True, False, False,  True,
         True,  True])]
```

Fig 4. Array received after comparing the reference and test images using the function *compare_faces* of *face_recognition* library. [9]

Now using ImageDraw and draw. *rectangles* function we can draw a box around the recognized face on the image which is available. We can also identify the location of individual features of the face with the help of *face_recognition. face_landmarks ()* method. These features act as coordinates to draw a box over the detected face in the image.

```
[{
    'chin': [(46, 47), (45, 54), (44, 62), (44, 69), (44,
    'left_eyebrow': [(51, 42), (54, 39), (58, 39), (63, 4
    'right_eyebrow': [(75, 44), (80, 44), (86, 44), (90,
    'nose_bridge': [(70, 48), (68, 52), (67, 56), (66, 60
    'nose_tip': [(60, 64), (62, 65), (65, 67), (68, 66),
    'left_eye': [(55, 47), (57, 45), (61, 46), (63, 48),
    'right_eye': [(77, 51), (80, 50), (84, 51), (86, 54),
    'top_lip': [(54, 75), (58, 72), (61, 72), (64, 73), (
    'bottom_lip': [(73, 80), (68, 81), (64, 81), (62, 80)
}]
```

Fig 5. Values of basic facial feature stored in *face_recognition. face_landmarks ()* [9]

VI. RESULTS

A. *Few of the Reference Images*

Fig 6. The above pictures have been used to train faces to our model.

B. *Test images*

Fig 7. Above pictures of people with face mask is fed for testing.

C. *Output*

Fig 8. Above is the detected and recognized pictures that were fed to the model.

Some pictures were not recognized due to several reasons such as lighting conditions, pixelated images etc. While testing we received 70% to 85% of accuracy (i.e., the program was able to recognize the person correctly), while in some situations the program was not able to recognize the person.

VII. Discussions

In this research, we were able to detect faces using PIL module and face_detection library available for Python. This specific module and library have been chosen because we face less difficulties compared to other methods since these inoperative methods were not able to detect a face because they needed the whole face, if a face covered with mask was imported, the model had hard time identifying the face, i.e., it needs all the facial features to recognize faces. Another possible problem to be faced is the resolution of the imported image itself, for example, if the image were to be blurred or has pixels too huge causing staircase effect or other seeable distortions, inaccurate values would be stored causing more complications. The image needs to readable by the model or else the model will not be able to get accurate data which further does not result in accurate outcome. Other challenge we had to come across was the unavailability of sufficient dataset with the required details.

Keeping dilemmas aside, this model or system gives accurate result even when applied with small constraints such as different angle, low resolution image, it is also able to recognize people with masks when the model has learnt the same person's face structure without their face covered.

VIII. Future Scope

As seen since the past year. The usage of mask has been made compulsory due to the COVID-19 pandemic for the ensuring health safety. Therefore, recognizing people wearing mask, while covering most of their facial features has never been of more prominent importance. In future face recognition while wearing masks can be implemented in smartphones and other devices where they use facial recognition for security reasons (Current versions of face recognition system in smartphones does not recognize a person's face if he/she is wearing a mask). This practice may or may not be continued but in some other circumstances it seems to be very vital in recognizing people in such aspect.

Currently our program works on a predefined reference dataset (i.e., if a person's face is detected which is not available in the given dataset the program provides the output *Unknown*). So, we would like to work on and understand how to identify people in such scenarios. We would also like to improve the functionality of our project by implementing emotion recognition while wearing masks.

The speed of recognizing a person is an important factor while performing facial recognition. Currently our program takes around 20 seconds to 30 seconds to recognize a person. So, we would like to implement features and work on our code for the same.

IX. Conclusion

In this research, we are able to detect and recognize faces of people with mask with the help of Python Image Library. Firstly we feed the model with the reference images, which are of people without mask and images taken at different angles. Using these images a consolidated data array is made with 128 specific points of the face, capturing all the unique details. Now when we feed the model to recognize the image of a person wearing the mask, the model compares the data array and makes an accurate guess. Considering our challenges faced and future developments in creation of new modules, the new techniques would definitely ease the work needed in analyzing and understanding an image. But this method restricts or limits our capability to use large number of sample pictures to study the features. This further could be modified to fix the limitations to load/ feed numerous reference images to the model, for its facial recognition training.

REFERENCES

[1] Qiao, W., Chen, B., Zhao, J. and Guo, S., 2021, March. Face Recognition Based on CD-RBM and BM-ILM. In Journal of Physics: Conference Series (Vol. 1802, No. 3, p. 032077). IOP Publishing.

[2] Li, X., Luo, J., Duan, C., Zhi, Y. and Yin, P., 2021, March. Real-Time Detection of Fatigue Driving Based on Face Recognition. In Journal of Physics: Conference Series (Vol. 1802, No. 2, p. 022044). IOP Publishing.

[3] Li, Y., Guo, K., Lu, Y. and Liu, L., 2021. Cropping and attention based approach for masked face recognition. Applied Intelligence, pp.1-14.

[4] Damer, N., Boutros, F., Süßmilch, M., Fang, M., Kirchbuchner, F. and Kuijper, A., 2021. Masked Face Recognition: Human vs. Machine. arXiv preprint arXiv:2103.01924.

[5] Loey, M., Manogaran, G., Taha, M.H.N. and Khalifa, N.E.M., 2021. A hybrid deep transfer learning model with machine learning methods for face mask detection in the era of the COVID-19 pandemic. Measurement, 167, p.108288.

[6] Maurya, P., Nayak, S. and Vijayvargiya, S., 2021. COVID-19 Face Mask Detection.

[7] Adusumalli, H., Kalyani, D., Sri, R.K., Pratapteja, M. and Rao, P.P., 2021, February. Face Mask Detection Using OpenCV. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 1304-1309). IEEE.

[8] Liu, G. and Zhang, Q., 2021. Mask Wearing Detection Algorithm Based on Improved Tiny YOLOv3. International Journal of Pattern Recognition and Artificial Intelligence, p.2155007.

[9] Vikas K. Solegaonkar, 2019. https://solegaonkar.github.io/FaceRecognition.html

[10] R. Ramachandran, S. Velan S and D. Imtiyaz Wadekar, "Statistical Comparison of COVID-19 Infections Based Upon the Food Habits/Diets in Countries Using RStudio," 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), 2021, pp. 1095-1101.

[11] Suganantham, S., Babu, C. and Raju, M., "A Quantitative Evaluation of Change Impact Reachability and Complexity Across Versions of Aspect Oriented Software", International Arab Journal of Information Technology (IAJIT), 14(1), 2017.