Face Recognition using Python and OpenCV

- A review

Paras Saini[1], Sanskar[2], Kritika Singh[3], Dashrath Nandan[4], Navneet Singh [5], Er. Kumud Sachdeva[6]

[1]Department of Computer Science, Chandigarh University, Mohali, India
21BCS11677@cuchd.in
[2]Department of Computer Science, Chandigarh University, Mohali, India
21BCS11572@cuchd.in
[3]Department of Computer Science, Chandigarh University, Mohali, India
21BCS11655@cuchd.in
[4]Department of Computer Science, Chandigarh University, Mohali, India
21BCS11716@cuchd.in
[5]Department of Computer Science, Chandigarh University, Mohali, India
21BCS11691@cuchd.in
[6]Department of Computer Science, Chandigarh University, Mohali, India
kumud.cse@cumail.in

---***---

Abstract – *Face detection is a computer means of determining the locations and sizes of human faces in video and photographs. It detects countenance but ignores objects like as buildings, trees, and objects. Human visual awareness is a hot topic in the computer vision world right now. Human facial localization and identification are frequently the first steps in applications such as video surveillance, human-computer interface, face recognition, and image management. Although a generic face picture is frequently accessible, identifying and monitoring human faces may be required for face recognition and/or countenance analysis. The subject of computer-based face recognition employing impartial facial data as an analysis is still mostly unknown. Given how people see faces and how they differ from verification robots, it should be interesting to see how machines favour distinct countenances rather than offering face recognition challenges. As a result, this work investigates the subject of face recognition utilizing inadequate facial information. The experiment is based on the use of Python in conjunction with OpenCV (Open Computer Vision) for accurate classification and identification of the face. Throughout this work, we will develop Face Detection and Tracking using Har characteristics*

***Key Words*:** Face detection, identification, Tracking, surveillance, verification, Face Recognition, Convolution neural network (CNN), Linear Binary Pattern (LBP), Graphical User Interface(GUI)

1. INTRODUCTION

Face recognition is a technique for determining a person's identification based on their particular face. Such systems can be found in photographs, films, and real-time devices. The goal here is to present a simple way of machine technology. With the use of such a technique, one may quickly recognize a person's face by using a dataset with a comparable matching look. The approach of detecting a person's face using Python and OpenCV in deep learning is the most efficient way. This strategy is beneficial in various settings, including the military, security, schools, colleges and universities, airplanes, banking, online web applications, gaming, etc. Automated face recognition systems have been developed to deal with these applications thanks to recent breakthroughs in automated face analysis, pattern recognition, and machine learning. On the one hand, identifying a face may be a natural behaviour since individuals recognize one other's faces. Face detection performance may be a major issue, so non-frontal face detection techniques are going to be discussed. Subspace modelling and learning dimension-based dimension reduction techniques are fundamental to current face recognition techniques. Another challenge in this area is to explore such sub-areas to uncover effective features and build robust classifications. Face recognition is characterized by both high accuracy and low penetration, so it's attracted the eye of researchers in a variety of fields, from psychology, and image processing to computer vision. The first step is face recognition in an acquired image, no matter the scale, and site.

2. LITERATURE SURVEY

There have been various types of face recognition throughout history. Face recognition system broadly has four modules namely

1. Face Detection
2. Face Alignment
3. Feature Extraction
4. Comparison of the test face image with the face image in the database.

The recognition process starts with identifying the face region in the given image. Tracking is performed to locate a human face if the input is a video sequence. Further from the detected frame, the face part is detected. The detected face region is aligned and adjusted as a major aspect of pre-processing. Features are extracted from the aligned face image using any of the feature extraction techniques and further matched with the features of face images from the gallery.

Bledsoe et al. [1] an American researcher, 1964, studied facial recognition computer programming. They imagine a semi-automatic method, where operators are asked to enter twenty computer measures, such as the size of the mouth or the eyes. In 1977, The system was improved by adding 21 additional markers (e.g., lip width, hair color). 1988: Artificial intelligence was introduced to develop previously used theoretical tools, which showed many weaknesses. Mathematics (“linear algebra”) was used to interpret images differently and find a way to simplify and manipulate them independently of human markers.

Wiskott et.al [2] use the Gabor wavelet transform to represent the local features. Lawrence had proposed a face recognition system that uses local image sample representation, Self-Organizing Map (SOM), and Convolution neural network (CNN).

Ahonen et al. [3] divided the Face image into sub-regions and calculate the Linear Binary Pattern (LBP) histograms. Later these LBP histograms are combined into a global histogram.

Fontaine et al [4]: The face detection using Viola-Jones face detector followed by which they detect 68 landmarks of the face using Dlib shape detector is discussed in their proposed work. They construct Delaunay triangulation mesh by placing equidistant points on the face border and detected landmarks. They perform pose alignment by applying affine transformation on the input image wrap mesh.

Wright et al [5] in 2009 stated that a lot of investigation has been done on the projection from high to low dimensions. But the problem is which features to use. Recently with compressed sensing, the feature space is no longer critical, what is important is that the dimension of the feature space is sufficiently large and that the sparse representation is correctly computed. This method outperformed other techniques for corrupted and occluded images.

Ding et al [6] present the difficulties in Pose Invariant Face recognition and review existing techniques. Ding et.al retrieve Multidirectional Multi-Level Dual-Cross Patterns (MDML-DCPs) from face images. MDML-DCPs encode the invariant characteristics of a face image into patterns that are robust to variations in faces belonging to the same class and have high discrimination against faces that belong to different classes.

Ganguly et al. [7] performed face recognition of the 3d face images in an unconstrained environment with variations in pose, occlusion, and lighting. The availability of additional information in the form of depth data in 3D face images is used to rotate. They used the Energy Range Face Image model to normalize terms of the pose variation and occlusion restoration.

Shroff et. al. [8], 2011, developed a face similarity measure that is invariant to pose, expression, and illumination. The similarity between the probe image with the images in the library is used to create the ordered list. Thus, each face image has its own ordered list which is referred to as the signature of the face. To determine the similarity of the probe image its signature is compared with the signatures of the images in the library. They used the data-driven approach using the Doppelganger list for each image from the library. The similarity of the pair can be determined by comparing the similarity of the list computed with the doppelganger list.

Rui Min et al. [9] 2011 worked on improving the recognition rate of faces that are occluded by facial accessories. In this work, the authors have considered sunglasses and a scarf as facial accessories. They developed a method where the presence of sunglass and scarfs are detected using Gabor wavelets, Support Vector Machines (SVM), and Principal Component Analysis (PCA). Once the face with occlusion of accessories is detected, the recognition is performed from the non-occluded region. This is done using block-based local binary patterns.

Zou et al. in 2012 [10] stated that the recognition performance of the existing methods on VLR face images will degrade dramatically. As most of the image details of the VLR face image are lost and it contains very minimal information. They categorize the algorithms into two approaches namely, example-based and maximum a posterior (MAP) based. They developed the relationship between the High-Resolution image and the VLR image, and later apply the relationship to recover the High-resolution images from VLR. Face images with a frontal view are used in the experiments

Li et al. [11], 2013, proposed a morphological graph model that describes the morphological structure of the occlusion. Incorporating the errors in the occluded part and non-occluded part, the authors proposed structured sparse error coding for face recognition from occlusion.

Ho et al. [12], 2013, used a variant of the belief propagation (BP) Algorithm and Markov Random Fields (MRFs) to generate a frontal view from a face image with the pose. The given probe image is first classified as a frontal or non-frontal view using SVM. The non-frontal probe image is split into grids with overlapping patches. The patches from the frontal view are created from the globally optimal set of local warps. The frontal face thus generated is used for face recognition.

Alyuz et al. [13], 2013, developed a 3-D face recognition system that is robust to occlusions. Missing data is handled using the subspace analysis technique. Nonoccluded patches are utilized for construction

zhu et al. [14], 2015, presented a facial recognition algorithm by morphing the input images into the model. A High-Fidelity Pose and Expression Normalization (HPEN) method is developed with a 3D Morphing Model (3DMM) to normalize expression and pose in the image to the frontal face. The model follows two steps, firstly a landmarking marching assumption is done to detect several feature points of the face on the given image. Secondly, the whole image is meshed into a 3D object to eliminate pose and expression variances using identity preserving 3D transformation.

Biswas et al. [15] extract SIFT descriptors from HR gallery and LR probe images and transform them into a space in which inter-Euclidean distances approximate the distances calculated for all the descriptors using HR frontal images. Multidimensional scaling is used to learn the desired transformation. They use SIFT-based descriptors as the input feature which are represented by the fiducial locations of the face image. The tensor analysis-based approach is used to predict rough locations of the facial landmarks and approximate pose. The scale factor between the HR gallery (60x55) and LR probe images is fixed at 3. Multiple Biometric Grand Challenge (MBGC) and FRGC datasets are used. The recognition performance using the proposed approach significantly improves over the SIFT combined with PCA features.

3. OBJECTIVES

- To write a program which stores new faces, their names, age and other information, that can later be used to mark that individual present on the list.
- To develop a face recognition system which, from the incoming image, finds a series of data of the same face in a set of training images in a database.
- We are to develop an efficient technique for processing and recognizing face images.
- To make the program display name and other details of the identified person on the screen.

4. PROBLEM STATEMENT

In this contemporary world, biometrics are very important. As the population is growing so is the difficulty maintaining unique id's whereas biometrics are different for every person and the reason why one should use facial recognition as one cannot change one's face other methods are not as reliable as this one.
A complete face recognition system includes face detection, face pre-processing and face recognition processes.
Therefore, it is necessary to extract the face region from the face detection process and separate the face from the background pattern, which provides the basis for the subsequent extraction of the face difference features.
The recent rise of the face based on the depth 8 of learning detection methods, compared to the traditional method not only shorten the time, and the accuracy is effectively improved.
Face recognition of the separated faces is a process of feature extraction and contrast identification of the normalized face images in order to obtain the identity of human faces in the images.

5. METHODOLOGY

The Processing of face recognition can be cut short in a few words and described as only 4 parts:

a. Recording the data: First of all, a face is recorded in system as image is encoded. Each frame is encoded

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

separately and even the exiting images are also encoded

b. Extraction: Then the encoded data is extracted from the image. Each image has a different encoded data.

c. Comparison: Then the extracted data is compared with any existing database image's encoded data.

d. Matching: The system decides if the features extracted from the new Samples may be a match or not a match.

These have been further explained below but first we need to understand what affects the image captured and the factors affecting face recognition. We will also understand what is Graphical user interface and Open CV, how it is used in our project.

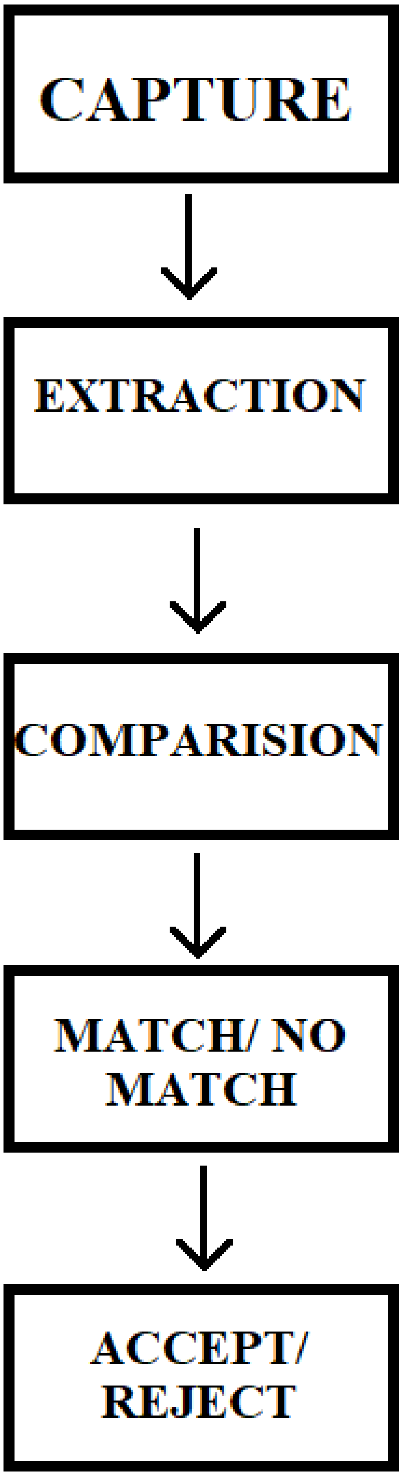

Figure 5.1 Basic Functioning of Face Recognition

When it comes to picture quality and video image, several aspects influence the system's accuracy. To standardize the photos, you offer to the face recognition system, you should use a variety of image pre-processing techniques. Because most face recognition algorithms are light-

sensitive, a person who has been taught to detect in a dark room will not be detected in a bright environment. Etc.
This issue "depends on the brightness" of the images as well as their pixels. In photographs, the face is highly stable (for example, the eyes are in the same pixel coordinates), and has a consistent form, rotation angle, hair, and cosmetics. Feelings (smiling, angry, etc.). The lighting situation (left or upward, etc.).
To keep things simple, the facial recognition system uses eigenfaces with grayscale photos. The study demonstrates how color photographs may be simply converted to grayscale (also known as 'grayscale') and then automatically apply histogram equation to the brightness and contrast of your facial images to identify face identification and recognition.
Standardization is a relatively easy process. Color Face Recognition (preferably with colour histogram fitting on HSV or any other colour space instead of RGB) or additional processing stages such as edge processing, contour detection, motion detection, and more can improve results. This code also resizes photos to a standard size, but it has the ability to modify the aspect ratio of the face. One way is to describe how to modify its size while maintaining the image's aspect ratio. OpenCV makes use of the Haar Cascade face detection package. When viewing a photograph from a disc file or another source.
The Face Detector evaluates the position of each picture in real-time and categorizes it as "face".
The faces in the image are smaller or larger, the classification goes around the image numerous times to find faces that meet the classification requirements.
We'll use the dlib facial recognition framework, which is a simple approach to arrange face evaluations. Dlib and face-recognition are the two most important libraries in the system. Python, being a sophisticated programming language and one of the most widely used programming languages in the world, has shown to provide the greatest results in face recognition and detection systems. With the aid of the Python programming language and OpenCV, face recognition and detection become quite simple and successful.

- **Graphical User Interface:**

The graphical user interface (GUI) is the platform that allows the user to interact with the system by providing inputs. Mobile phones, video players, gaming, and other devices all employ graphical user interfaces.
In every software application, we may create visual composition and the temporal behaviour of the GUI, as well as programming in the fields of human-computer interaction. The user interface for this project will be heavily dependent on the training and testing phases, allowing for picture collection and training.

Gary Bradski proposed the idea of OpenCV, a multi-level framework that could accomplish tasks. OpenCV comes with a variety of useful features and functions right out of the box. OpenCV assists in the recognition of a person's frontal face and generates XML documents for a variety of locations, including bodily parts. In the process of recognising systems, deep learning has recently emerged. As a result, deep learning and face recognition operate as deep metric learning systems when used together. In a nutshell, deep learning in face detection and identification will primarily focus on two areas: taking a solidary input image or any other relevant image, and providing the best outputs. We would be using dlib facial recognition framework that would be the easy way to organize the face evaluation. The two main significant libraries used in the system are dlib and face-recognition.

We must first create the datasets in order to build this system. Different operations in the face recognition system will take place as the image quality improves. The jobs are completed using the python query "python encode faces.py." The input will come from the dataset that is received in the "encodings.py" file.
Precision formatting will be used in the system, with face embedding for each face. Second, a file called "recognize faces images.py" will include all of the necessary procedures and approaches for identifying a person's face from a particular image in the dataset. The python command "python recognize faces image.py-encodings" will run the specified file. We can resize or turn the image for approximate with the goal for getting the desired output.
It is also necessary to create a file named Attendence.csv, this file will contain names, time and date of the captured face if a number is assigned instead of a name then that will be displayed instead of a name.

6. CONCLUSION

Face detection is used in a variety of areas, including security and human tracking. It is dependent on improving identity performance; there are several areas where improvements may be made, some of which are simple to implement. Color processing, edge detection, and other features can be added. By capturing more and more images of each individual, you may enhance the face recognition accuracy and make better decisions on the dataset, especially at diverse angles and lighting circumstances. If you can't take any more photos, there are a few basic strategies you may utilize to make additional training photos by combining existing images: You may increase the size of your past training photos by making mirror duplicates of your facial photographs. You may translate or resize photographs of your face, as well as rotate them somewhat, to produce numerous different training images that are less sensitive to precise settings.

REFERENCE

[1] W. W. Bledsoe, "Man-machine facial recognition", *Rep. PRi*, vol. 22, 1966.
[2] L. Wiskott, "Phantom Faces for Face Analysis", Pattern Recognition, vol. 30, no. 6, pp. 837-846, 1996.
[3] T. Ahonen, A. Hadid and M. Pietikinen, "Face Recognition with Local Binary Patterns", Proc. Eighth European Conf. Computer Vision, pp. 469-481, 2004.
[4] X. Fontaine, R. Achanta and S. Süsstrunk, "Face recognition in realworld images", 2017 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), pp. 1482-1486, March 2017.
[5] J. Wright, A. Yang, A. Ganesh, S. Sastry and Y. Ma, "Robust Face Recognition via Sparse Representation", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 31, no. 2, pp. 210-227, Feb. 2009.
[6] C. Ding, J. Choi, D. Tao and L. S. Davis, "Multi-directional multi-level dual-cross patterns for robust face recognition" in , 2014
[7] Suranjan Ganguly (Department of Computer Science and Engineering, Jadavpur University, Kolkata, India), International Journal of System Dynamics Applications (IJSDA)
[8] Schroff, F., Treibitz, T., Kriegman, D. and Belongie, S., 2011, November. Pose, illumination and expression invariant pairwise face-similarity measure via doppelgänger list comparison. In 2011 International Conference on Computer Vision (pp. 2494--2501). IEEE
[9] R. Min, A. D'angelo and J.-L. Dugelay, "Efficient scarf detection prior to face recognition", EUSIPCO 2010 18th European Signal Processing Conference, pp. 259-263, Aug. 2010.
[10] Zou, W.W.W.; Yuen, P. Very Low Resolution Face Recognition Problem Image Processing, IEEE Transactions on, Jan 2012, 21, 327-240
[11] Xioao-Xin li, Structured Sparse Error Coding for Face Recognition With Occlusion, 2013.
[12] H.T. Ho, Pose-Invariant Face Recognition Using Markov Random Fields, Volume: 22, Issue: 4, April 2013
[13] Nese Alyuz, 3-D Face Recognition Under Occlusion Using Masked Projection, Volume: 8, Issue: 5, May 2013
[14] M. Shen, B. Ma, L. Zhu, R. Mijumbi, X. Du, and J. Hu. 2018. Cloud-Based Approximate Constrained Shortest Distance Queries Over Encrypted Graphs With Privacy Protection. IEEE Transactions on Information Forensics and Security 13, 4 (April 2018)
[15] S. Biswas, G. Aggarwal and P.J. Flynn, "Face Recognition in Low-Resolution Videos Using Learning-Based Likelihood Measurement Model", Proc. Int'l Joint Conf. Biometrics, 2011.