FACE DETECTION AND RECOGNITION USING OPENCV AND PYTHON

***Hameem Mohammed, ** Vipin Vasu A. V**

**P. G. Student, **Associate Professor,*

Dept. Of Computer Engineering,

College of Engineering, Thiruvanandhapuram, Kerala.

ABSTRACT:

Face recognition is one of the many marvels that AI research has brought to the world. Many techies are interested in this topic because they want to have a basic understanding of how things work. Let's delve into the subject to see how it works. This paper describes a method for extracting a photo of the attributes in order to determine someone's face. The Cascade object detection method is included in the proposed method. In the face of any image, this method is efficient and effective.

Keywords: *Python, OpenCV, Deep Learning, Face detection, Face recognition*

INTRODUCTION:

Face recognition is one of the most important biometric authentication techniques. It is the most fascinating and successful application of Pattern Recognition and Image Analysis. For the face detection method, the input image is usually rescaled. The primary functions of a face recognition system are verification and identification. Verification is defined as a 1:1 match between images of a face and images of a template face whose identity is being claimed. Identification is a 1: N problem that compares the image of the query face to all template images in a face database.

This research paper describes the methods and applications of selected face recognition and their application in relation to face recognition, as well as the need for this study to interact in the further development of face recognition. [1]

PROBLEM DEFINITION:

The following is a common formulation of the overall description of the face recognition problem (in computer vision): detecting or verifying one or more elements of a scene, and identifying one or more people within the scene using a database of faces.

Face recognition usually consists of two steps:-

1. Face Detection:

Whereas the photograph is searched for the person's face, the image is processed to chop and extract the person's face for easy identification.

2. Face Recognition:

The discovered and processed face of the person is compared to a face database, from which the person is determined. Since 2002, OpenCV, Open Cell's open source framework, has made face recognition easier and more reliable. The structure includes a built-in face detector that detects 90-95% of candid photos of people looking at the camera. However, it is often difficult to identify an individual's face when viewing it from a distance, and sometimes a 3D head estimation is required. Furthermore, a lack of proper image brightness exacerbates the problem of identifying the face, or the difference in shadow on the face, or the image is blurred or the person is wearing glasses. Facial recognition is far less reliable than face recognition, with an average accuracy of 30-70%. Face recognition has been a thriving field of study since the 1990s, but it is still a long way from being a reliable method of consumer authentication. Every year, more and more technologies are developed. Although the eigenface technique is considered the only accurate method of face recognition, many other (more complex) techniques or combinations of various techniques are more accurate. [2]

OpenCV:

OpenCV has the advantage of being a cross-platform framework, supporting Windows, Linux, and, most recently, Mac OS X. OpenCV has so many capabilities that it can be intimidating at first. The key to getting good results with OpenCV is to understand how it works. Fortunately, only a few people choose to get started. The functionality of OpenCV is used in many modules for facial recognition.

PROPOSED SOLUTION:

When it comes to image quality and video image, there are numerous factors that influence the system's accuracy. To standardise the images you present to the face recognition system, use a variety of image pre-processing techniques. Most facial recognition algorithms are light sensitive, so if a person is trained to detect in a dark room, it will not detect them in a bright room. Etc. This issue is "dependent on the brightness" and the pixel count of the images. In photographs, the face is very stable (for example, the eyes are in the same pixel coordinates), with a consistent shape, rotation angle, hair, and makeup. Emotion (smiling, angry etc) (smiling, angry etc). The lighting situation (left or upward, etc.). Before performing face recognition, it is critical to use a good image pre-processing filter. [3]

OpenCV employs a face detector library known as Haar CasCade. The Face Detector examines the location of each image and classifies it as "face" or "face" when viewing a photo from a disc fille or live video. Because the faces in the image are smaller or larger, the classification moves around the image many times in order to find faces that meet the classification criteria. This can be time-consuming, but thanks to the algorithmic techniques described in the sidebar, the classification is extremely fast, even when applied to multiple scales. The classifier determines how to classify each image based on data stored in the XML file. OpenCV Download includes one of four flavours of XML data with profile faces for frontal facial

recognition. It requires three non-face XML files: one for full body (pedestrian) identification, one for upper and lower body, and one for both. [4]

PHASES IN FACE RECOGNITION:

a) Face Detection:

It is critical to detect facial features in an image in order to recognise a face. During this phase, we must also add a few random images to the dataset.

b) Training the dataset:

The dataset must be trained using OpenCV Recognizer in the second step. This is possible with the OpenCv function. The training Data.yml file will be saved in the "trainer/" directory as a result.

C) Face Recognition:

The final step is to capture a brand new face on camera, and if this person's face has previously been captured and trained, the recognizer will make a "prediction," returning its id and an index, indicating how confident the recognizer is with this match. Recognizer for cv2. Predict () accepts a captured portion of the face to be analysed as a parameter and returns its Matched Image.

It displays the Image Id as well as the recognizer's level of confidence in this Match. It's worth noting that the arrogance index will return "zero" if it's deemed an ideal match. It predicts a Face in the final step. We will also display the name over the image, along with the likely id and the amount of "probability" that the match is correct ("probability" = 100 confidence index). If this is not the case, a "unknown" label is placed on the face.

OBJECTIVES:

1. A method for detecting and recognising human faces with OpenCV and Python
2. To develop a system that will use the computer's camera or a system that will detect and recognise a person's face using openCV and Python.
3. To investigate the various stages of face detection and recognition using openCV.

REVIEW OF LITERATURE:

This section provides a basic overview of the major techniques used in the face recognition system, which are mostly applicable to the human's front face. Neural networks, hidden Markov models, geometric face matching, and template matching are among the methods used. Eigenface is one of the most widely used methods in face recognition and detection, which are referred to in mathematical terms as the principle components. The eigenvectors are arranged to represent various amounts of variation in the faces.

Face recognition and detection systems make extensive use of neural networks. In face recognition, an ANN (artificial neural network) with a single layer was used, demonstrating adaptability in critical face recognition systems. Face recognition is accomplished through the use of a double layer of WISARD in neural networks.

Another option for face recognition is graph matching. Graph matching, which is performed by optimising a matching function, can be used to formulate both object and face recognition.

A. Krizhevsky et al proposed a model that employs a pre-trained neural network. It's a CNN model that competed in the ILSVRC-ImageNet large scale visual recognition challenge; they just tweaked it for their face detection problem. The model was trained using the Annotated Facial Landmarks in the Wildm dataset, which contains 21K images with 24K face annotations. [5]

S. Matuska et al. compared the performance of OpenCV and Matlab. The basic image processing algorithm is presented, as well as the time consumption in OpenCV and Matlab. It has been demonstrated that OpenCV is up to 30 times faster than Matlab and can be up to 100 times faster for the Erosion algorithm. [6]

Caricatures [Brennan 1985; Bruce 1988; Perkins 1975]: A caricature is formally defined [Perkins 1975] as "a symbol that exaggerates measurements relative to any measure that varies from one person to another." Thus, the length of a nose varies from person to person and could be used as a symbol in caricaturing someone, but not the number of ears. A standard caricature algorithm [Brennan 1985] can be applied to various qualities of image data (line drawings and photographs). Line-drawing caricatures do not contain as much information as photographs, but they do capture the important characteristics of a face; experiments on non-ordinary faces comparing the usefulness of line-drawing caricatures and unexaggerated line drawings decisively favour the former [Bruce 1988]. [7-9]

Distinctiveness [Bruce et al. 1994]: Studies show that distinctive faces are better remembered and recognised than typical faces. When deciding whether an object is a face or not, recognising an atypical face takes longer than recognising a typical face. This could be explained by the use of different detection and identification mechanisms. [10]

Face recognition and movement [O'Toole and colleagues, 2002; Bruce et al., 1998; Knight and Johnston, 1997]: According to a recent study [Knight and Johnston 1997], famous faces are easier to recognise when shown in moving sequences rather than still photographs. This observation has been extended to demonstrate that movement aids in the recognition of familiar faces shown under a variety of degradations—negated, inverted, or thresholded [Bruce et al. 1998]. Even more intriguing is the observation that there appears to be a benefit from movement even when the information content in the moving and static comparison conditions is equated. Experiments with unfamiliar faces, on the other hand, show that viewing animated rather than static sequences provides no additional benefit. [11-12]

RESEARCH METHODOLOGY:

Gary Bradski proposed the OpenCV concept, which could operate on a multi-level framework. OpenCV has a number of important capabilities as well as utilities that are visible from the start. OpenCV assists in recognising a person's frontal face and also generates XML documents for various areas such as body parts.

Python, a very powerful programming language and one of the most widely used programming languages in the world, has proven to provide the best results in face recognition and detection systems. Face recognition and detection become very simple and fruitful when combined with the Python programming language and OpenCV.

Books, educational and development journals, government papers, and print and online reference resources were just a few of the secondary sources we used to learn about the design, implementation, and impact of Face Detection and Recognition using OpenCV and Python.

RESULT AND DISCUSSION:

When the image quality becomes favourable different procedures will take place in the face recognition system the tasks are performed using the python queries "python encode faces.py". The dataset received in "encodings.py" will be used as input. Precision formatting will occur in the system, with face embedding for each face. Second, a file called "recognize faces images.py" will contain all of the necessary methods and techniques for identifying a person's face from a dataset image. The python command "python recognize faces image.py-encodings" will run the given file.

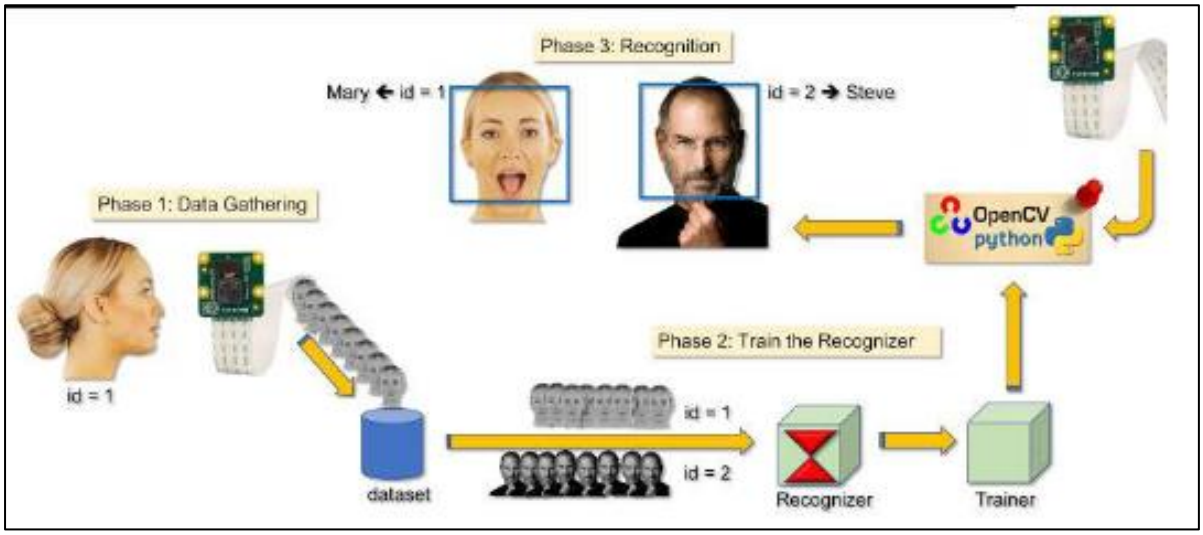


Figure 1: face recognition system design using python and OpenCV.

We can resize or rotate the image to get a rough estimate of the desired output. The current classifier, in conjunction with the OpenCV libraries, will improve the outcome or results of the face recognition system. [13]

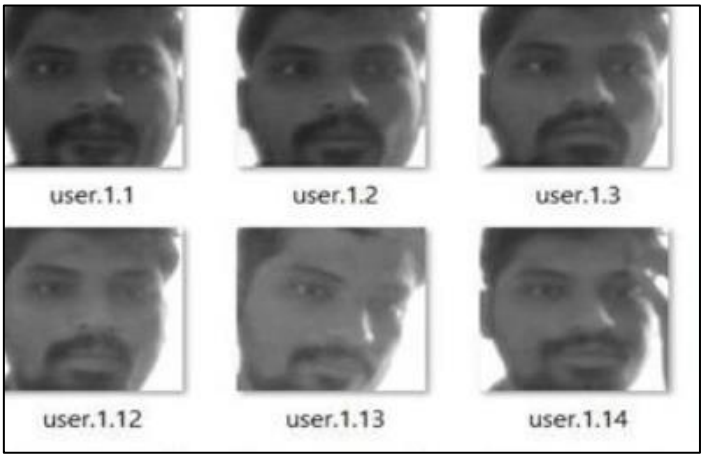
2. **Using Phases in Face Recognition, we analysis the following result in stepwise**

Step1: Face Detection:

The user's facial features are detected and saved in the dataset.

Step2: Training the dataset:

The most critical step is to train the captured data. The data is trained using the Open Cv recognizer and the OpenCv function, which produces the "trainer.yml" file, which is saved in the "trainer/" directory.



Step3: Face Recognition:

The final step is to test the functionality of the face detection. Face features are captured by the camera in this case; if the captured features are already present in the dataset, the recognizer will predict them and return the person's name and id (or whatever is given when registered to reflect when the data is matched). If not, the user must first register his or her face in order to record the features. [14]

CONCLUSION:

Face recognition systems are now associated with many leading technological companies and industries, making face recognition work easier. The use of Python programming and OpenCV makes it a simpler and more useful tool or system that anyone can create based on their needs. The proposed system discussed in this project will be beneficial to many people because it is user friendly and cost effective. As a result, the face recognition system can be designed for a variety of purposes using Python and OpenCV.

REFERENCE:

1. K. T. Talele, S. Kadam, A. Tikare, Efficient Face Detection using Adaboost, “IJCA Proc on International Conference in Computational Intelligence”, 2012.
2. M. A. Turk and A.P. Pentland, Face recognition using eigenfaces, “Proceedings of the IEEE”, 586-591, 1991.
3. P. Viola & M. Jones (2001), “Rapid Object Detection using a Boosted Cascade of Simple Feature”, Conference on Computer Vision and Pattern Recognition. IEEE Press, Pp. 511–518.
4. Open Source Computer Vision Library Reference Manual-intel
5. A.Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Advances in neural information processing systems, 2012, pp. 1097-1105
6. S. Matuska, R. Hudec, and M. Benco, “The Comparison of CPU Time Consumption for Image Processing Algorithm in Matlab and OpenCV,” pp. 75–78, 2012
7. BRENNAN, S. E. 1985. The caricature generator. Leonardo, 18, 170–178.
8. BRUCE , V. 1988. Recognizing faces, Lawrence Erlbaum Associates, London, U.K
9. PERKINS, D. 1975. A definition of caricature and recognition. Stud. Anthro. Vis. Commun. 2, 1–24
10. BRUCE , V., B URTON, M., AND D ENCH, N. 1994. What’s distinctive about a distinctive face? Quart. J. Exp. Psych. 47A, 119–141

11. O'T OOLE, A. J., ROARK, D., AND A BDI, H. 2002. Recognitizing moving faces. A psychological and neural synthesis. Trends Cogn. Sci. 6, 261–266.
12. K NIGHT, B. AND J OHNSTON, A. 1997. The role of movement in face recognition. Vis. Cog. 4, 265–274.
13. Face Detection and Tracking using OpenCV. S.V.Viraktamath, Mukund Katti, Aditya Khatawkar, Pavan Kulkarni. 3, s.l.: SIJ, July-August 2013, The Standard International Journals (The SIJ) , Vol. 1, pp. 45-50. ISSN: 2321 – 2403
14. Face Detection in Real Time supported HOG. N. J. Wang,S. C. Chang and P. J. Chou. Taipei, Taiwan: IEEE, DOI:10.1109/ISPACS.2012.6473506, 2012. International Symposium on Intelligent Signal Processing and Communications Systems. pp. 333-337. ISBN: 978-1-4673- 5081-5.