Real Time Face Detection and Recognition using OpenCV and Python

By:

Tarang Agarwal

tarangleo@gmail,com

Abstract –*This research study demonstrates how to detect and recognize human faces using OpenCV and Python, both of which are deep learning tools. This report explains how to use numerous libraries in OpenCV and Python to detect the face using deep learning, and it's an essential element of the computer science field. This article will provide a proposed method that will aid in the real-time detection of human faces. This implementation can be utilized on a variety of platforms, including computers and smartphones, as well as a variety of software applications*

Key Words: Python, OpenCV, Deep Learning, Face detection, etc.

1. INTRODUCTION

Face recognition is a method for determining a person's identification based on their unique facial characteristics. Photos, films, and real-time machines may all benefit from such systems. The purpose of this essay is to present a more straightforward and straightforward approach to machine technology. With the use of such technology, a person's face may be simply detected using a dataset with a similar matching look. The approach of detecting a person's face with the aid of python and OpenCV in deep learning is the most efficient. This strategy may be used in a variety of settings, including the military, security, schools, colleges and universities, airplanes, banking, web applications, and gaming etc. this system uses powerful python algorithm through which the detection and recognition of face is very easy and efficient.

1.1 Motivation

Biometrics, which is used for authentication and makes work simpler, is the most helpful field in which facial recognition is vital. Face recognition is one of the most widely used technologies or systems, with the ability to perform tasks such as having records provided in by the dataset in many areas such as school and college attendance systems, it can also be useful in catching thieves or terrorists, and it can be useful in the security of common people and the country's much-needed security areas. Face recognition may be used by the government to validate voter lists, discover missing people, find the population or census, and speed up the immigration process. It can help protect Ecommerce from internet frauds and is widely used in medical and healthcare. This creates a high demand for a real-time facial recognition system, which can be used for a variety of purposes by the public and the government.
There would be ease in a variety of activities if such superb solutions were available.

1.2 Problem Statement

The major goal or goal of this work is to build or construct a system that will utilize the computer's camera to identify and recognize a person's face or an individual's face using the OpenCV tool called Open Face and the Python programming language in the deep learning domain.

2. Literature survey

This section provides a general overview of the key approaches used in face recognition systems, which mostly pertain to the human front face. Neural networks, hidden Markov models, geometric face matching, and template matching are among the approaches used.

In mathematical terms, Eigenface is one of the most extensively used approaches in face identification and detection, and it is referred to as the principal components. The eigenvectors are arranged in a certain sequence to indicate various degrees of variation in the faces.

Face identification and detection systems frequently employ neural networks. In face recognition, an ANN (artificial neural network) with a single layer was utilized, demonstrating adaptability in critical face recognition systems. A double layer of WISARD in neural networks is used for facial verification.

Another approach for facial recognition is graph matching. Graph matching, which is achieved by optimizing a matching function, may be used to develop both object and face recognition.

Hidden Markov Models are a method of stochastic modelling of nonstationary vector time series based on the HMM model applied to human face identification, in which the faces are separated into sections such as the eyes, nose, and ears, among others. Face recognition and proper matching are 87 percent accurate because it always selects the best and most appropriate face detection option from a stored dataset. Alternatively, the applicable model exposes the face's identity.

The technique of geometrical feature matching is based on the geometrical forms of the face. Face detection and

identification systems can use the geometrical face configuration since it has a large enough dataset. This is one of the most widely used face recognition and detection methods. This system appears to produce good outcomes.

Template matching is one of the ways in which the test picture is represented as a two-dimensional array of values that can be compared with a single template representing the entire face using Euclidean distance. This approach may also depict an individual face by combining many face templates from various angles.

3. Methodologies

Gary Bradski proposed the idea of OpenCV, which has the capacity to work on a multi-level framework. OpenCV comes with a variety of powerful features and functions right out of the box. OpenCV assists in identifying a person's frontal face and provides XML documents for a variety of locations, including bodily parts.
Deep learning has recently become popular in the field of recognition systems. As a result, deep learning and face recognition work together to form deep metric learning systems. 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 or results for the image of the picture. We'll use the dlib facial recognition framework, which is a simple approach to structure 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.

3.1 Need of an automated system

Due to the general growing demand for systems that can assist in areas such as surveillance and security, this type of individual authentication can no longer be done using simple hand-crafted methods, necessitating the development of automated systems that can quickly correct errors and process human face recognition. When labor is done by machines, it can complete jobs effectively in a short amount of time and eliminates the significant errors that humans make. Face detection may be made easier using a real-time GUI-based face recognition system, which can be done in a variety of ways.

3.2 Minimum Requirements

The Python, OpenCV, and the relevant dataset would constitute the software's minimal prerequisites. The minimum hardware requirements are an Intel i3 or higher processor and a four-core CPU. Windows 10 operating systems will suffice, and 8GB of random-access memory will be required. A PC or laptop with an active internet connection, as well as a scanner, are required on the user's end. This project's user interface will be heavily influenced by the training and testing phases, allowing for picture collection and training.

4. Proposed Arrangement for system design

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 approximate the intended outcome by resizing or turning the image. The current classifier, in conjunction with the OpenCV libraries, will improve the face recognition system's outcome or results.

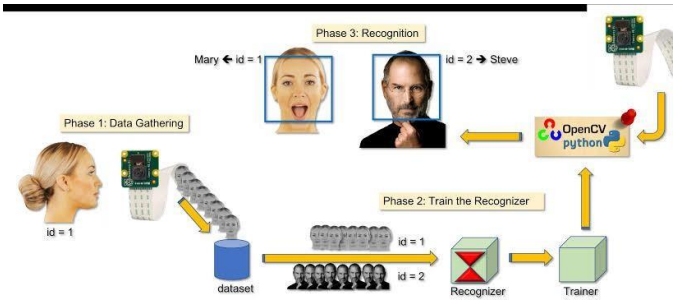

Figure 1: face recognition system design using python and OpenCV.

5. Advantages and Disadvantages

The Faster processing, identity automation, breach of privacy, massive data storage, best results, enhanced security, real-time face recognition of students in schools and colleges, employees at corporate offices, smartphone unlock, and many other benefits of the face recognition system can be found in everyday life.

The costing, or funding, of this system, very good high-definition cameras are required, poor image quality may limit the effectiveness of this system, image size matters because it is difficult to recognize the face in small images, face angles can limit the face recognition reliability, and massive storage is required for this system to work effectively are just a few of the disadvantages.

6. Conclusions

Face recognition technologies are now affiliated with a number of leading technical firms and sectors, making face recognition work easier. The usage of Python programming with OpenCV makes it a simpler and more useful tool or system that anybody may create according to their needs. The suggested system mentioned in this project would be beneficial to a large number of people since it is user-friendly and cost-effective. As a result, a face recognition system may be created for a variety of uses using Python and OpenCV.

7. Acknowledgement

I would like to sincerely thank my mentor Mr Prateek Narang and Manu s Pillai for guiding me throughout this project work also would like to thank Coding Blocks for allowing me to perform this project work.

8. References

[1] Face Detection in Real Time Based on 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.

[2] Face Recognition Homepage,

http://www.face﷯rec.org/a lgorithms/

[3] Seeing with OpenCV, Article, http://www.cognotics.com/opencv/servo _2007_series/part_1/index.html

[4] Python with deep learning,

https://www.tutorialspoint.c

[5] CS229 Notes - Andrew NG (Simple & Easy to understand)