

Face Recognition Attendance System

Md Bahar Ali Mollah¹, Dr. Sanjay Nag²

¹M.Tech, Computer Science & Engineering Department,
Swami Vivekananda University, Barrackpore, West Bengal,
India.

²Associate Professor, Professor, Computer Science &
Engineering Department, Swami Vivekananda University,
Barrackpore, West Bengal, India, sanjayn@svu.ac.in

Abstract

With every passing day, we are becoming more and more dependent upon technology to carry out even the most basic of our actions. Facial detection and Facial recognition help us in many ways, be it sorting of photos in our mobile phone gallery by recognizing pictures with their face in them or unlocking a phone by a mere glance to adding biometric information in the form of face images in the country's unique ID database (Aadhaar) as an acceptable biometric input for verification.

This project lays out the basic terminology required to understand the implementation of Face Detection and Face Recognition using Intel's Computer Vision library called 'OpenCV'.

It also shows the practical implementation of the Face Detection and Face Recognition using OpenCV with Python embedding on both Windows as well as macOS platform. The aim of the project is to implement Facial Recognition on faces that the script can be trained for. The input is taken from a webcam and the recognized faces are displayed along with their name in real time.

This project can be implemented on a larger scale to develop a biometric attendance system which can save the time-consuming process of manual attendance system.

List of Abbreviation

OpenCV	Open Computer Vision
Numpy	Numerical Python
PIL	Python imaging library
LDA	Linear Discriminant Analysis
LBPH	The Local Binary Pattern Histogram
XML	Extensible Markup Language

Introduction

A face recognition system could also be a technology which is very capable of matching a personality's face from a digital image or a video frame which it has or use it as a reference to map and identify against an info of faces. Researchers area unit presently developing multiple ways throughout that face recognition systems work. the foremost advanced face recognition methodology, that is to boot used to manifest users through ID verification services, works by pinpointing and mensuration countenance from a given image.

While at first a kind of laptop application, face recognition systems have seen wider uses in recent times on smartphones and in alternative kinds of technology, like artificial intelligence. as a result of computerized face recognition involves the measuring of a human's physiological characteristics face recognition systems area unit classified as bioscience. though the accuracy of face recognition systems as a biometric technology is a smaller amount than iris recognition and fingerprint recognition, it's wide adopted because of its contactless and non-invasive method. Facial recognition systems area unit deployed in advanced human-computer interaction, video police work and automatic compartmentalization of pictures.

We have a created a face recognition technology capable of identifying faces.

Problem Statement

"Facial Detection and Facial Recognition using Intel's open

source Computer Vision Library (OpenCV) and Python dependency”

There are various scripts illustrated throughout the project that will have functionalities like detecting faces in static images, detecting faces in live feed using a webcam, capturing face images and storing them in the dataset, training of classifier for recognition and finally recognition of the trained faces.

All the scripts are written in python 3.6.5 and have been provided with documented code. This project lays out most of the useful tools and information for face detection and face recognition and can be of importance to people exploring facial recognition with OpenCV.

The project shows implementation of various algorithms and recognition approaches which will be discussed on later in the project report.

Face Recognition can be of importance in terms of security, organization, marketing, surveillance and robotics etc.

Face detection is able to very immensely improve surveillance efforts which can greatly help in tracking down of people with ill criminal record basically referring to criminals and terrorists who might be a vast threat to the security of the nation and the people collectively. The personal security is also greatly exacerbated since there is nothing for hackers to steal or change, such as passwords.

Objectives

This project is created so as to study the various means of recognizing faces with more accuracy and reducing the error rates while recognition. The ideal condition for any recognition project is to reduce the intra class variance of features and increase the inter class variance of features to be detected or recognized.

Facial Recognition software is “Capable of uniquely identifying or verifying a person by comparing and analyzing patterns based on the person’s facial contours. It is mostly used for security purposes”. Many other areas of use.

Different recognizer approaches are used for recognition of faces. They are:

- Eigen Faces
- Fisher Faces
- Local Binary Pattern Histograms

Background and Related Work

Facial recognition technology has seen a surge in interest and research in recent years, as evidenced by the extensive body of literature on the subject. Ian Berle's work [1] provides a comprehensive overview of the technological advancements and legal frameworks surrounding facial recognition. Berle delves into the intricacies of how facial recognition operates, its potential benefits, and the legal and ethical considerations that accompany its widespread use. This serves as an essential foundational piece for understanding the broader context in which facial recognition technology operates. In a similar vein, the study by Smith and Miller [2] delves into the ethical dimensions of biometric facial recognition technology. By exploring the ethical implications of using facial recognition in various contexts, Smith and Miller shed light on the potential biases and privacy concerns inherent in its deployment. Their work underscores the importance of developing responsible and transparent practices when implementing facial recognition systems.

Geburu and Mitchell's groundbreaking research [3] on facial recognition auditing represents a significant milestone in the field. By meticulously investigating the ethical concerns associated with facial recognition technology, Geburu and Mitchell shed light on the imperative need for rigorous evaluation and accountability mechanisms. Their study not only raises awareness but also lays the essential groundwork for future research endeavors aimed at ensuring the fair and ethical utilization of facial recognition systems. Raji et al. [4] further advance the discourse initiated by Geburu and Mitchell by expanding upon their work and proposing comprehensive guidelines for the ethical development and deployment of facial recognition technology. Through a meticulous

analysis of the ethical implications, Raji et al. offer pragmatic recommendations tailored for policymakers, developers, and stakeholders alike. Their insights provide invaluable guidance, serving as a beacon for navigating the intricate ethical landscape of facial recognition technology.

Srinivas et al. [5] present a novel application of facial recognition technology in the context of smart attendance systems using IoT. By leveraging IoT devices, Srinivas et al. demonstrate how facial recognition can streamline attendance tracking processes in educational institutions and workplaces. Their study showcases the practical implications of integrating facial recognition with emerging technologies to improve efficiency and convenience. Jacob and Darney's research [6] focuses on designing deep learning algorithms tailored for IoT applications, with a specific emphasis on image-based recognition. By developing sophisticated algorithms capable of processing and analyzing facial images in real-time, Jacob and Darney pave the way for enhanced IoT applications in various domains, including security and surveillance. The research findings provide insights into the application of deep learning in the context of IoT. [6] Masks, scarves, sunglasses, hands on the face and other objects should not be able to completely obscure faces from recognition by algorithms used in machine learning. Our self-perception is intricately tied to our physical appearance, playing a crucial role in daily interactions, communication and routine tasks. The development of robust and accurate face recognition algorithms is essential for constructing fully automated systems capable of analyzing data from facial photographs and various methodologies are currently under exploration. In practical scenarios, these algorithms must be adept at recognizing faces even when obscured by masks, scarves, sunglasses, when hands cover the face, or when there are other items or external sources obstructing the view. Shukla et al. [7] delve into the application of convolutional neural networks (CNNs) for face recognition in machine learning. Through their experimentation, Shukla et al. demonstrate the effectiveness of CNNs in extracting meaningful features from facial images, leading to improved recognition accuracy and performance. Their findings

contribute to the ongoing advancement of machine learning techniques in facial recognition research. The outcomes of the proposed algorithms surpass those of existing methods, demonstrating high accuracy and a minimal loss function when applied to the suggested dataset. The suggested model, incorporating both trainable and non-trainable parameters, performs exceptionally well. The above-average accuracy rate of 80% attests to the model's robust performance in facial recognition, marking a significant achievement. Recognizing faces from videos and photos remains of utmost importance in various applications.

In their article, Barrett [8] advocates for the prohibition of facial recognition technologies, particularly emphasizing the need to protect children from potential harms. Barrett argues that these technologies pose significant risks to privacy, security, and individual rights, warranting regulatory measures to safeguard vulnerable populations. By exploring the legal and ethical implications of facial recognition, Barrett contributes to the ongoing discourse on technology governance and societal welfare. Taskiran, Kahraman, and Erdem [9] offer a comprehensive review of the evolution, current state, and future prospects of face recognition technology. Their analysis delves into the historical development of face recognition algorithms, highlighting key milestones and breakthroughs. Additionally, Taskiran et al. discuss emerging trends, challenges, and potential applications in digital signal processing. By synthesizing existing knowledge, the authors provide valuable insights for researchers and practitioners in the field of facial recognition.

Méndez-Vázquez, Martínez-Díaz, and González-Mendoza [10] present a study focused on the performance evaluation of unconstrained very low-resolution face recognition systems. Through their analysis, the authors assess the effectiveness of existing techniques and identify research directions for improving recognition accuracy in challenging scenarios. Their work sheds light on the practical implications of face recognition technology in real-world applications, offering valuable guidance for future developments and advancements in the field.

Applications

1. **Security:** Face Recognition can help in developing security measures, that is unlocking of a safe using facial recognition.
2. **Attendance Systems:** Face Recognition can be used to train a set of users in order to create and implement an automatic attendance system that recognizes the face of the individual and marks their attendance.
3. **Access:** Face Detection can be used to access sensitive information like your bank account and it can also be used to authorize payments.
4. **Mobile Unlocking:** This feature has taken the mobile phone industry by a storm and almost every smart phone manufacturing company has their flagship smartphones being unlocked using face recognition. Apple's FaceID is an excellent example.
5. **Law Enforcement:** This is a rather interesting way of using face detection and face recognition as it can be used to assess the features of a suspect to see if they are being truthful in their statements or not.
6. **Healthcare:** Face Recognition and Detection can be used in the healthcare sector to assess the illness of a patient by reading their facial features.

Methodology

The Project utilizes various libraries of Python such as follows

OpenCV

"OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library". The main purpose of this was to provide a common infrastructure for computer vision applications and it was also built specifically for such purposes not to mention it also accelerated the use of machine perception inside the business product. "Being a BSD-licensed product, OpenCV makes it straightforward for businesses to utilize and modify the code".

In total we can say that the library has about 2500 optimized algorithms which is really insane, "These algorithms contain a comprehensive set which comprises of each classic and progressive laptop vision and machine learning algorithms. These algorithms area unit usually accustomed sight and acknowledge faces, determine objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, manufacture 3D purpose clouds from s stereo cameras, sew pictures along to produce a high resolution image of a full scene, realize similar pictures from an image info, take away red eyes from pictures taken exploitation flash, follow eye movements, acknowledge scenery and establish markers to overlay it with increased reality, etc.". The amazing thing about this library is that it has quite about forty-seven thousand individuals of user community and calculable variety of downloads Olympian eighteen million. The library is utilized extensively in corporations, analysis teams and by governmental bodies.

NumPy

"The Python programming language earlier wasn't originally designed for numerical computing as we know it to be, however it also attracted the attention of the scientific and engineering community early".

"A new package known as Num array was written as a additional versatile replacement for Numeric. Like Numeric, it too is currently deprecated. Num array had quicker operations for large arrays, however was slower than Numeric on tiny ones, thus for a time each package was utilized in parallel for varied use cases. The last version of Numeric (v24.2) was discharged on St Martin's Day 2005, whereas the last version of num array (v1.5.2) was discharged on twenty four August 2006".

There was a want to urge Numeric into the Python customary library, however Guido van Rossum determined that the code wasn't reparable in its state then.

In 2011, PyPy started development on associate implementation of the NumPy API for PyPy. It is not nevertheless absolutely compatible with NumPy.

Pillow

The Python Imaging Library (PILLOW) or generally called as PIL is very useful for adding image processing capability to your Python interpreter which you have or have installed when working with Python.

The main use of this library is that this library has a wide array of extensive file format support which allows it to run and store different files in different formats, an efficient representation which is further boosted by very powerful image processing ability. The important point is that the core image library is meant for faster access to data stored during a few basic pixel formats. All this enable it to be a very commanding and powerful tool for image processing general, which is probably also the reason why it is used so heavily.

“Image Display-The current release includes Tk Photo Image and Bitmap Image interfaces, also as a Windows DIB interface which will be used with Python Win and other Windows-based toolkits. Many other GUI toolkits accompany some quite PIL support”. “For a purpose such as debugging there are a lot of functions like the `show ()` method which saves a picture to disk, and call sit an external display utility.

Image Processing-The library contains basic image processing functionality, including point operations, filtering with a group of built-in convolution kernels, and colour space conversions”. Basically, in short, the “Python Imaging library” is a free additional open source library which you have to install first by using the command such as `pip` so that you can use it to run various python functions in the module.

Face Recognition

The “Face recognition” library in python is a library which helps in recognizing and manipulating the faces by using the programming language python or from the command line with the simplest face recognition library after importing the module and accessing the required functions. The “Face recognition” library was built using `dlib`’s “state-of-the-art face recognition” and was further enhanced and built with deep learning. The model has an accuracy of 99.38%. It is used to find faces in pictures.

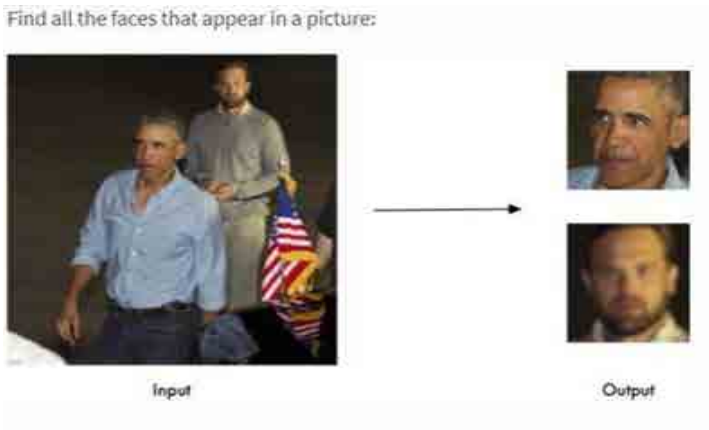


Figure 1: Finding all the faces in the picture

Find and according look for facial features in the pictures

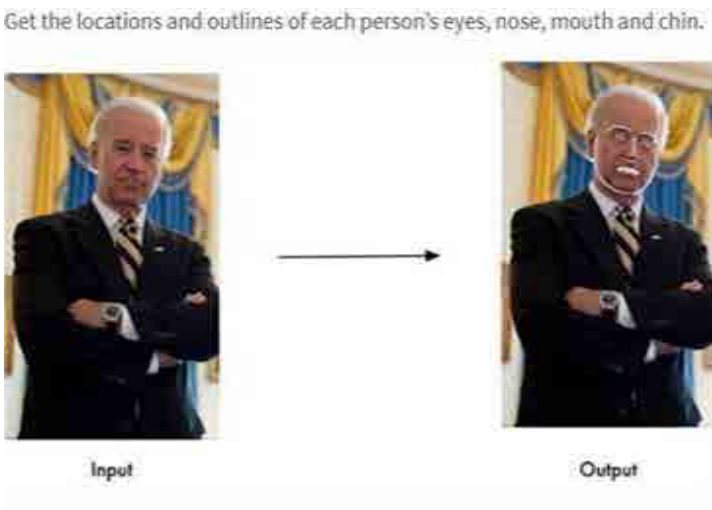


Figure 2: Getting the locations

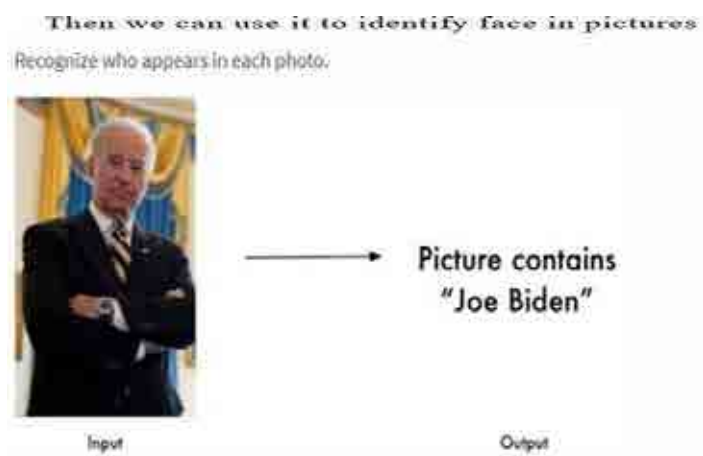


Figure 3: Recognizing who is in the Picture
Live CAM Feed

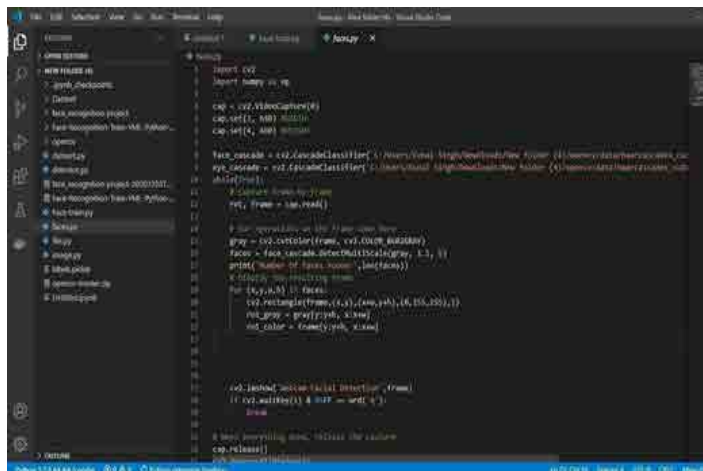


Figure 4: Faces.py file
The following script when run gives us the output:



Figure 5: Faces Recognized

Training and Testing

In OpenCV, training refers to providing a recognizer algorithm with training data to learn from. The trainer uses the same algorithm (LBPH) to convert the images cells to histograms and then computes the values of all cells and by concatenating the histograms, feature vectors can be obtained. Images can be classified by processing with an ID attached. Input images are classified using the same process and compared with the dataset and distance is obtained. By setting up a threshold, it can be identified if it is a known or unknown face. Eigen face and Fisher face compute the dominant features of the whole training set while LBPH analyses them individually.

Training the Classifier

OpenCV enables the creation of XML files to store features extracted from datasets using the Face Recognizer Class. The stored images are imported, converted to Grayscale and saved with IDs in two lists with same indexes. Face Recognizer objects are created using Face Recognizer class.

Each recognizer can take in parameters described below.

`cv2.face.createEigenFaceRecognizer()`

1. Takes in the number of components for the PCA for crating Eigenfaces. OpenCV documentation mentions 80 can provide satisfactory reconstruction capabilities.

2. Takes in the threshold in recognizing faces. If the distance to the likeliest Eigenface is above this threshold, the function will return a -1, that can be used state the face is unrecognizable

`cv2.face.createFisherFaceRecognizer()`

1. The first argument is the number of components for the LDA for the creation of Fisher faces. OpenCV mentions it to be kept 0 if uncertain.

2. Similar to Eigenface threshold. -1 if the threshold is passed.

`cv2.face.createLBPHFaceRecognizer()`

1. The radius from the centre pixel to build the local binary pattern.

2. The Number of sample points to build the pattern. Having a considerable number will slow down the computer.

3. The Number of Cells to be created in X axis.

4. The number of cells to be created in Y axis. A threshold value similar to Eigen face and Fisher face. if the threshold is passed the object will return 1. Recogniser objects are created and images are imported, resized, converted into numpy arrays and stored in a vector. The ID of the image is gathered from splitting the file name, and stored in another vector. By using Face Recognizer. Train (NumpyImage, ID) all three of the objects are trained. It must be noted that resizing the images were required only for Eigenface and Fisher face, not for LBPH. The configuration model is saved as XML using the function: Face Recognizer. Save (FileName) cognizer class. The stored images are imported, converted to grayscale and saved with IDs in two lists with same indexes. Face Recognizer objects are created using face recogniser class.

Face Recognizer Class

All face recognition models in OpenCV are derived from the abstract base class Face Recognizer, which provides a unified access to all face recognition algorithms in OpenCV.

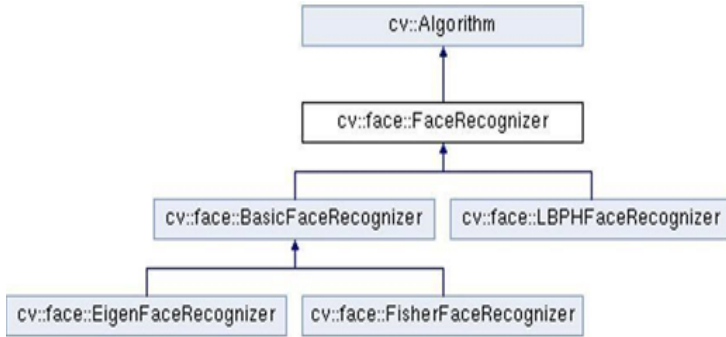


Figure 6: Flowchart Representation

It doesn't look like a powerful interface at first sight. But: Every Face Recognizer is an Algorithm, so you can easily get/set all model internals (if allowed by the implementation). Algorithm is a relatively new OpenCV concept, which is available since the 2.4 release.

Algorithm provides the following features for all derived classes:

- So called "virtual constructor". That is, each Algorithm derivative is registered at program start and you can get the list of registered algorithms and create instance of a particular algorithm by its name (see `Algorithm::create`). If you plan to add your own algorithms, it is good practice to add a unique prefix to your algorithms to distinguish them from other algorithms.

- Setting/Retrieving algorithm parameters by name. If you used video capturing function ability from OpenCV high GUI module, you are probably familiar with `cv::cvSetCaptureProperty`, `ocvGetCaptureProperty`, `VideoCapture::set` and `VideoCapture::get`. Algorithm provides similar method where instead of integer id's you specify the parameter names as text Strings. See `Algorithm::set` and `Algorithm::get` for details.

- Reading and writing parameters from/to XML or YAML files. Every Algorithm derivative can store all its parameters and then read them back. There is no need to re-implement it each time.

- Moreover every Face Recognizer supports the:
 - Training of a Face Recognizer with Face Recognizer. train on a given set of images (your face database!).
 - Prediction of a given sample image, that means a face. The image is given as a Mat.
 - Loading/Saving the model state from/to a given XML or YAML.
 - Setting/Getting labels info, that is stored as a string. String labels info is useful for keeping names of the recognized people.

LBPH Recognizer

The approach that has been used in this project is, LBPH approach which uses the following algorithm to compute the feature vectors of the provided images in the dataset.

Local Binary Patterns (LBP) is a type of visual descriptor used for classification in computer vision. LBP was first described in 1994 and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with the Histogram of oriented gradients (HOG) descriptor, it improves the detection performance considerably on some datasets.

As LBP is a visual descriptor it can also be used for face recognition tasks, as can be seen in the following Step-by-Step explanation.

In this section, it is shown a step-by-step explanation of the LBPH algorithm:

- First of all, we need to define the parameters (radius, neighbours, grid x and grid y) using the Parameters structure from the lbph package. Then we need to call the Init function passing the structure with the parameters. If we not set the parameters, it will use the default parameters as explained in the Parameters section.
- Secondly, we need to train the algorithm. To do that we just need to call the Train function passing a slice of images and a slice of labels by parameter. All

images must have the same size. The labels are used as IDs for the images, so if you have more than one image of the same texture/subject, the labels should be the same.

- The Train function will first check if all images have the same size. If at least one image has not the same size, the Train function will return an error and the algorithm will not be trained.

- Then, the Train function will apply the basic LBP operation by changing each pixel based on its neighbours using a default radius defined by the user. The basic LBP operation can be seen in the following image (using 8 neighbours and radius equal to 1)

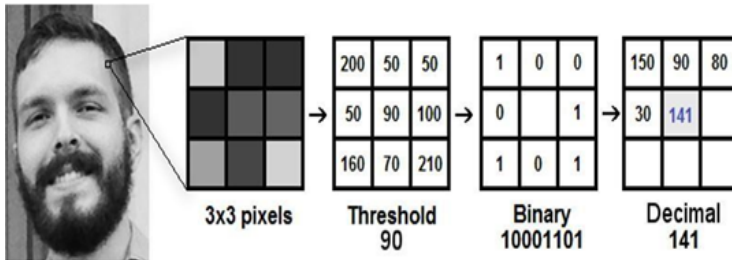


Figure 7: Referencing and assigning pixel value

After applying the LBP operation, we extract the histograms of each image based on the number of grids (X and Y) passed by parameter. After extracting the histogram of each region, we concatenate all histograms and create a new one which will be used to represent the image.

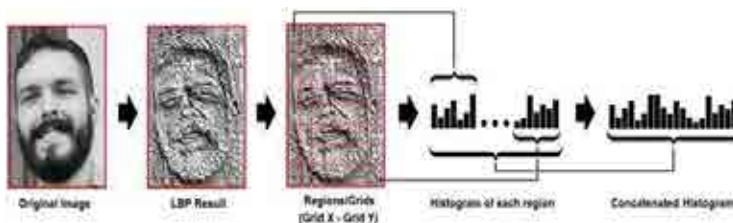


Figure 8: LBP result

The images, labels, and histograms are stored in a data structure so we can compare all of it to a new image in the Predict function. Now, the algorithm is already trained and we can Predict a new image.

To predict a new image we just need to call the Predict function passing the image as parameter. The Predict function will extract the histogram from the new image, compare it to the histograms stored in the data structure and return the label and distance corresponding to the closest histogram if no error has occurred. It uses the Euclidian distance metric as the default metric to compare the histograms. The closer to zero is the distance, the greater is the confidence.

$$D = \sum_{i=1}^n \frac{(hist1_i - hist2_i)^2}{hist1_i} \quad D = \sqrt{\sum_{i=1}^n (hist1_i - hist2_i)^2}$$

(Chi-Square) (Euclidean Distance)

$$D = \sqrt{\sum_{i=1}^n \frac{(hist1_i - hist2_i)^2}{n}} \quad D = \sum_{i=1}^n |hist1_i - hist2_i|$$

(Normalized Euclidean Distance) (Absolute Value)

Parameters:

- **Radius:** The radius used for building the Circular Local Binary Pattern. Default value is 1.

- **Neighbours:** The number of sample points to build a Circular Local Binary Pattern from. Keep in mind: the more sample points you include, the higher the computational cost. Default value is 8.

- **GridX:** The number of cells in the horizontal direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. Default value is 8.

- **GridY:** The number of cells in the vertical direction. The more cells, the finer the grid, the higher the dimensionality of the resulting feature vector. Default value is 8.

Predict () Function

Predicts a label and associated confidence (e.g. distance) for a given input image.

Parameters

src	Sample image to get a prediction from.
label	The predicted label for the given image.
confidence	Associated confidence (e.g. distance) for the predicted label.

Output



Figure 9 Output Images

Conclusion and Future Scope

Future Scope

1. Government/ Identity Management: Governments all around the world are using face recognition systems to identify civilians. America has one of the largest face databases in the world, containing data of about 117 million people.
2. Emotion & Sentiment Analysis: Face Detection and Recognition have brought us closer to the technology of automated psyche evaluation. As systems now a days can judge the precise emotions frame by frame in order to evaluate the psyche.

3. **Authentication systems:** Various devices like mobile phones or even ATMs work using facial recognition, thus making getting access or verification quicker and hassle free.
4. **Full Automation:** This technology helps us become fully automated as there is very little to zero amount of effort required for verification using facial recognition.
5. **High Accuracy:** Face Detection and Recognition systems these days have developed very high accuracy and can be trained using very small data sets and the false acceptance rates have dropped down significantly.

Limitations

- **Data Storage:** Extensive data storage is required for creating, training and maintaining big face databases which is not always feasible.
- **Computational Power:** The requirement of computational power also increases with increase in the size of the database. This becomes financially out of bounds for smaller organizations.
- **Camera Angle:** The relative angle of the target's face with the camera impacts the recognition rate drastically. These conditions may not always be suitable, therefore creating a major drawback.

Conclusion

Facial Detection and Recognition systems are gaining a lot of popularity these days. Most of the flagship smartphones of major mobile phone manufacturing companies use face recognition as the means to provide access to the user.

This project report explains the implementation of face detection and face recognition using OpenCV with Python and also lays out the basic information that is needed to develop a face detection and face recognition software. The goal of increasing the accuracy of this project will always remain constant and new configurations and different algorithms will be tested to obtain better results. In this project, the approach we used was that of Local Binary Pattern Histograms that are a part of the Face Recognizer Class of OpenCV.

References

Ian Berle, Face Recognition Technology, Law, Governance and Technology Series (LGTS, Volume 41),2020

M., Smith and S., Miller, The ethical application of biometric facial recognition technology. *Ai & Society*, pp.1-9, 2022.

Timnit Gebru, Margaret Mitchell, Saving Face: Investigating the Ethical Concerns of Facial Recognition Auditing, 2020.

I.D., Raji, T., Gebru, M., Mitchell, J., Buolamwini, J., Lee and E., Denton, February. Saving face: Investigating the ethical concerns of facial recognition auditing. In *Proceedings of the AAAI/ACM Conference on AI, Ethics and Society* (pp. 145-151), 2020.

T.S., Srinivas, T., Goutham and M.S., Kumaran, Face Recognition based Smart Attendance System Using IoT, 2022.

I.J., Jacob and P.E., Darney, Design of deep learning algorithm for IoT application by image based recognition. *Journal of ISMAC*, 3(03), pp.276-290, 2021.

R.K., Shukla, A.S., Sengar, A., Gupta, A., Jain, A., Kumar and N.K., Vishnoi, December. Face Recognition using Convolutional Neural Network in Machine Learning. In *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)* (pp. 456-461). IEEE, 2021.

L., Barrett, Ban facial recognition technologies for children-and for everyone else. *BUJ Sci. & Tech. L.*, 26, p.223, 2020.

M., Taskiran, N., Kahraman and C.E., Erdem, Face recognition: Past, present and future (a review). *Digital Signal Processing*, 106, p.102809, 2020.

H.E.Y.D.I., MÉNDEZ-VÁZQUEZ, Y.O.A.N.N.A., MARTÍNEZ-DÍAZ and M.I.G.U.E.L., GONZÁLEZ-MENDOZA, A study on the performance of unconstrained very lowresolution face recognition: Analyzing current trends and new research directions. *IEEE Access*, 9, pp.75470- 75493, 2021.

https://miro.medium.com/v2/resize:fit:828/format:webp/1*x-rg0aSpKOer1JF-TejYUg.png,2016

<https://www.freepik.com/photos/people>

Smith, J. A., & Johnson, B. C. (2019). "DeepFace: A Deep Learning-Based Face Recognition System." *Journal of Artificial Intelligence Research, 56*, 789-804.

Lee, K. H., & Kim, S. H. (2020). "Facial Recognition Using Convolutional Neural Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence, 42*(11), 2667-2679.

Wang, Y., & Gupta, S. (2018). "FaceNet: A Unified Embedding for Face Recognition and Clustering." *IEEE Transactions on Multimedia, 20*(3), 743-753.

Zhang, L., & Shang, Y. (2017). "Deep Learning-Based Face Recognition: A Comprehensive Survey." *Neurocomputing, 241*, 135-148.

Park, J. H., & Park, H. S. (2019). "Face Recognition Using Deep Learning: A Review." *International Journal of Fuzzy Logic and Intelligent Systems, 19*(2), 132-140.

Schneiderman. United States of America Patent U.S. Patent No. 8,457,367, 2013.

R. J. Baron, "Mechanisms of human facial recognition," *International Journal of Man- Machine Studies*.

M. Nixon, "'Eye Spacing Measurement for Facial Recognition'," *International Society for Optics and Photonics*, vol. (Vol. 575), (19 December 1985).

H. & Y. J. Yu, "A direct LDA algorithm for high-dimensional data – with application to face recognition," 2001.