Proceedings of the 8th International Conference on Communication and Electronics Systems (ICCES 2023)
IEEE Xplore Part Number: CFP23AWO-ART; ISBN: 979-8-3503-9663-8

2023 8th International Conference on Communication and Electronics Systems (ICCES) | 979-8-3503-9663-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCES57224.2023.10192743

Machine Learning Approach for a Novel Facial Recognition System

Balanageshwara S[1],
Assistant Professor/ECE
Moodlakatte Institute of Technology, Kundapura
Kundapura, Karnataka – 576217, India
nageshwar.boss@gmail.com

Dr. Abdul Kareem[2],
Professor/ECE
Moodlakatte Institute of Technology, Kundapura
Kundapura, Karnataka – 576217, India
afthabakareem@gmail.com

Mr. Varuna Kumara[3],
Assistant Professor/ECE,
Moodlakatte Institute of Technology, Kundapura
Kundapura, Karnataka – 576217, India
vkumarg.24@gmail.com

***Abstract*- A facial recognition system can be developed using a machine learning approach that involves data collection, preprocessing, feature extraction, model training, evaluation and testing, and deployment. The system can be trained on a large dataset of facial images using techniques such as PCA, LBP, or CNNs for feature extraction and SVM, Random Forest, or Neural Networks for model training. The performance of the system can be evaluated using a test set, and the system can be deployed in real-world scenarios. However, it is crucial to consider the ethical and privacy implications of facial recognition technology and implement appropriate safeguards to prevent misuse. The Eigenface, Fisherface, and LBPH (Local Binary Patterns Histogram) algorithms are three popular techniques for face recognition in the OpenCV library. This work evaluates the performance of each algorithm on a specific dataset to determine which algorithm is the most appropriate for this application.**

Keywords – Machine learning, Face recognition technique, Algorithm.

I. INTRODUCTION

Facial recognition is an area that, despite having existed for decades, has gained prominence in recent years because it is more present in people's daily lives. This technology is applied, for example, as a way to unlock cell phones, access control in private areas, security systems and search for criminals in public areas [1]. So, for facial recognition to have public acceptance, its applications depend on models with the lowest possible error rate. However, the use of masks has become a limiting factor. In the current scenario, the deadly COVID-19 pandemic has changed people's daily lives due to the ease with which the virus spreads through the air and surfaces [2]. Therefore, the use of face masks has become an essential and proven means of protection in people's daily lives. However, facial recognition models were directly affected by the use of masks by reducing detectable facial areas, such as the mouth and nose. In addition to the reduction in the ability to classify faces with a mask, another factor that indicates the need for studies in this area is the use of this technology as a form of prevention when applied in the biometric area. Commonly used means for biometrics, such as the use of electronic cards and fingerprints, are not recommended in this pandemic because they require contact with potentially contaminated surfaces. In this way, facial biometrics gains prominence as an alternative for identifying people and combating identity fraud in a safe way for health. Therefore, this work seeks to use current methodologies to circumvent the problem applied to facial recognition. For this, suggestions will be presented for adapting existing models and ways to create new classifiers [3].

II. LITERATURE REVIEW

The first account of an attempt at facial recognition by a computer is by Bledsoe (1966), in which he tried to find the relationship of a photograph with records of photographs in a book. In this study, Bledsoe was able to highlight different problems that are recurrent until today in facial recognition, such as: variability of rotation and tilt of the head, intensity and angle of lighting, facial expression and aging. Years later, the authors were able to show that images of faces could be efficiently represented with Principal Component Analysis (PCA). The PCA is able to evaluate the distance between faces in a face space and thus determine similar faces [4-5].

With the optimization in the representation of a facial image, the need for more data for application in real scenarios arose. So there was a bet that facial recognition would be a more powerful way to identify individuals when compared to fingerprint. This bet encouraged the creation of the first large-scale database of faces, called Face Recognition Technology. Then, with advances in facial recognition, it was realized that a database with more varied and natural images of everyday life was needed for facial recognition models to be applied in the real world. To solve this, databases with images collected from the internet began to be produced and this exponential increase in data enabled the popularization of deep learning [6-7].

Thus, the AlexNet architecture was a milestone in the area of computer vision and the use of deep convolutional neural networks. This title was earned for being the first architecture in

convolutional neural networks to reach first place in the ImageNet base challenge when compared to traditional machine learning methodologies. This database contained 1000 classes with more than a million images (Figure 1) and the AlexNet network reduced the classification error by 10%. So, the use of convolutional neural networks for image classification has become the most used option until the present day [8].

Another milestone in facial recognition was the work [9] Created by researchers at Facebook, it was able to surpass the human capacity for face recognition for the first time. This was possible with a nine-layer convolutional network trained on the largest face dataset at the time, with four million facial images belonging to more than 4,000 identities, all pulled from the social network itself. This architecture achieved an accuracy of 97.35%, a reduction of more than 27% of the error compared to the previous state of the art.

Literature [10] explores the same database proposed in this work but for a larger number of classes ranging from 50, 60, 70 and 100 names of people. Another relationship with the current work is the use of convolutional neural networks with transfer learning. As a pre-processing methodology, the base images were rotated until the eyes were aligned parallel to the horizontal edges of the image. In addition, the regions of the mask were removed, leaving only the regions above it for training. Thus, the network obtained 91.3% accuracy for 60 classes of real images and 88.9% accuracy for 60 classes of artificially generated images.

The problem statement/research gap is the need to develop more robust and efficient systems that can handle unconstrained environments and work with low-quality and noisy images, as well as the need for more comprehensive and diverse datasets and efficient algorithms for real-time face recognition.

Incorporating multimodal biometric data has the potential to significantly improve the accuracy and robustness of face recognition systems, particularly in challenging environments. It could be a promising direction for future research in this field.

III. DEVELOPED PROTOTYPE

LBPH (Local Binary Patterns Histogram) is a widely used algorithm for face recognition. It works by extracting features from an image of a face based on the distribution of local binary patterns (LBP). LBP is a texture descriptor that describes the local patterns in an image by comparing the intensity values of a central pixel with its surrounding pixels. LBPH works by dividing the face image into small regions and computing the LBP histogram for each region. The histograms are then concatenated to create a single feature vector that represents the face.

To develop a prototype for evaluating the performance between the OpenCV Eigenface, Fisherface, and LBPH libraries, capable of identifying faces using global representations of the facial image, the following steps can be taken:

- Data Collection: Collect a dataset of facial images for training and testing the prototype. Ensure that the dataset is diverse and representative of the population that the prototype will be used on.
- Data Preprocessing: Perform preprocessing steps on the dataset such as cropping the faces, resizing them, and normalizing the pixel values to prepare them for the machine learning algorithms.
- Feature Extraction: Extract the features from the preprocessed dataset using Eigenface, Fisherface, and LBPH algorithms. Each algorithm will generate a feature vector that represents the unique characteristics of the faces.
- Model Training: Train the machine learning models on the extracted features using a classification algorithm such as Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), or Random Forests. The goal is to create a model that can accurately classify faces into different identities.
- Model Evaluation: Evaluate the performance of the models on a separate dataset that it hasn't seen before. Use metrics such as accuracy, precision, recall, and F1-score to measure the performance of the models.
- Prototype Development: Based on the evaluation, develop a prototype that uses the best performing algorithm for facial recognition.
- Prototype Testing: Test the prototype on a real-world scenario, making sure to take into account ethical and legal considerations.
- Continuous Improvement: Monitor the performance of the prototype and continuously improve it by collecting more data, fine-tuning the model, and integrating new algorithms and techniques.

By following these steps, a prototype can be developed that uses global representations of facial images to identify faces using the Eigenface, Fisherface, and LBPH libraries in OpenCV. The prototype can be continuously improved to achieve better performance in real-world scenarios.

A. Tools used:

The following tools were used for the development of the software:

a) PyCharm Community Edition version 2019.3

Developed by the company JetBrains, the PyCharm ide provides greater productivity in Python programming language projects. In addition to the free Community edition, the company also offers the full Professional edition with support for HTML, JS and SQL.

b) Webcam with 0.92megapixel static image resolution, integrated into the DELL Inspiron model 5458 notebook.

B. Language used:

As for the language, in this software prototype the following programming language was used:

i. Python:

Python is an object-oriented, cross-platform, versatile and popular programming language. It was created in the early 1990s by Guido Van Rossum at the Center for Mathematics and Computer Science in the Netherlands (CWI).

C. OpenCV Algorithms

The algorithms that will be used for the extraction of facial features and already implemented in the OpenCV library, will be the Eigenface, Fisherface and LBPH (Local Binary Patterns Histograms) methods.

i. Eigenface:

The human face has parts that are not essential in the facial recognition process. A person can be recognized by specific features of the face , such as the shape of the eyes, nose, forehead, etc.

Eigenface uses the same principle. The algorithm analyzes the training images of the faces, keeps all the important and necessary features for recognition and discards the rest. This process is known as Principal Components Analysis (PCA). PCA is a widely used method for dimensionality reduction in face recognition. It works by transforming the original high-dimensional face images into a lower-dimensional space that preserves the most important variations in the data. In face recognition, PCA is often used to extract a set of eigenfaces that represent the most important features of the face images.

Thus, in the Eigenface method, the algorithm will extract the main characteristics of the set of training images and, according to the variation of the pixel values, will generate several eigenface, or “ghost” photos. The sum of this variation of values will generate an average face. Based on the linear combination of the components of the different eigenfaces together with the middle face, it will be possible to reconstruct the original face.

The figure below represents the eigenfaces extracted from the faces of a database, as well as the average face that will be the basis for the reconstruction of the different faces.

Fig 1 - Middle face and eigenfaces

The Eigenface algorithm classifies the faces based on the calculation of the nearest neighbor pixel (KNN) distances of the original image with the various eigenfaces. Faces detected with a distance value outside the confidence limit (threshold) will be considered unknown.

To analyze the performance of Eigenfaces, the process involves preparing data, extracting features, training a recognition model, evaluating performance, and comparing with other methods.

ii. Fisherface

PCA is not an ideal method to perform the separation between face classes, as it considers lighting as an important factor in recognition. Thus, faces of the same person that have a change in lighting can be considered as the face of a different person. Fisherface does not capture lighting variations as obviously as the Eigenfaces method.

Unlike Eigenface, which uses PCA, Fisherface employs the LDA (Linear Discriminant Analysis) method for face recognition.

The technique used in the LDA was developed by the mathematician R.A. Fisher in the 1930s and aims to find a linear combination of features that separates two or more classes. The objective is to maximize the ratio of the cross-class hash matrix to the within-class hash matrix.

iii. LBPH

The Local Binary Patterns Histogram (LBPH) algorithm is a texture-based feature extraction method commonly used in facial recognition systems. It works by extracting texture features from an input image and generating a feature vector

that represents the texture pattern distribution within the image.

One of the first computational steps in LBPH is to create an intermediate image that describes the original image in a better way, highlighting facial features. From a grayscale image, a part of this image can be obtained as a 3x3 pixel window, as shown in the image below:

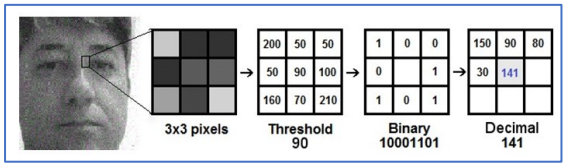

Fig 2 - Application of the LBP operation.

The LBPH algorithm does not create an original image. Instead, it extracts features from an input image and generates a feature vector that represents the texture pattern distribution within the image. In LBPH, the input image is divided into small regions called cells, and a binary code is generated for each cell based on the comparison of the intensity of the central pixel with its surrounding neighbors. This binary code is then converted to a decimal number, which represents the texture pattern of that cell. LBPH then creates a histogram of the texture patterns within a region of the face. This region is defined by a group of neighboring cells, which can be thought of as a patch. The histogram counts how many times each texture pattern appears within the patch. After normalizing the histograms and concatenating them, the resulting feature vector represents the texture pattern distribution of the entire image. This feature vector can then be used for face recognition. So, LBPH does not create an original image. Instead, it extracts texture features from the input image and creates a feature vector that can be used for facial recognition (Fig. 13).

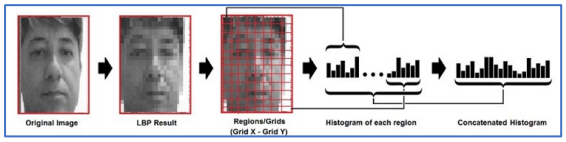

Fig 3 - Histograms of Local Binary Patterns

D. Prototype Validation

To improve the performance of a face recognition system trained on a large dataset of facial images, techniques such as data augmentation, feature combination, fine-tuning, hyperparameter optimization, ensemble learning, transfer learning, and hardware acceleration can be used.

In order to analyze the performance of the Eigenfaces, Fisherfaces and LBPH algorithms, the image bank of the Yale Face package, widely disseminated on the internet, was used, as shown in the figure below:

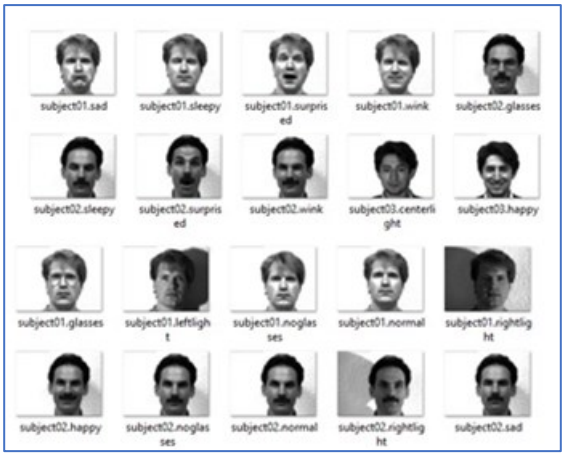

Fig 4 - Yale face pack

135 images were used for training, 9 images of each individual, totalling 15 people. In the training phase, the parameters PCA, Threshold, Radius, Grid X and Grid Y were used. To obtain a better accuracy in the results, the values were changed to higher and lower than the default value of the parameters.

The accuracy achieved for model training can vary depending on the quality of the dataset, the choice of features, and the model architecture.

i. Eigenfaces

The table below shows the results obtained in tests with the Eigenfaces algorithm. The first column represents the percentage of correct answers obtained in the tests, represented by the simple arithmetic mean of the number of correct answers with the sum of the number of faces. The second column represents the confidence value, basically represented by the distance value to find the most similar face in the data set, that is, the smaller the distance value, the greater the level of certainty that the algorithm has about the recognition. Finally, the third column has the values of the parameters used.

Table 1- Eigenface algorithm test results

Hits%	Confidence
68.97	6859.395
48.97	3610.585
58.97	4548.845
62.30	4825.075
65.63	5196.835
65.63	5196.835
55.63	3718.785
38.97	3995.805

32.30	3313.575
32.30	3313.575
5.63	1468.995
68.97	6859.395
68.97	6859.395
68.97	6859.395
68.97	6859.395
38.97	3995.805
68.97	6451.735

According to the table above, in the tests with the Eigenfaces algorithm, the best result of hits obtained was 70%, with parameter values: PCA numbers ranging from 134 to 150 and Threshold 100,000.

ii. Fisherfaces

In tests with the Fisherfaces algorithm, the best result of hits obtained was 96.66%, with parameter values: PCA numbers 13 and Threshold 8,000.

Table 2- Fisherfaces algorithm test results

Hits %	Confidence
85.63	2258.095
55.63	402.845
68.97	654.615
38.97	846.405
85.63	2258.095
85.63	2258.095
85.63	2258.095
92.30	2147.885
92.30	2147.885
92.30	2147.885
92.30	2147.885
95.63	1972.875
95.63	1893.855
95.63	1809.115
92.30	1644.125
92.30	1644.125
92.30	1644.125

iii. LBPH

In tests with the LBPH algorithm, the best result of correct answers obtained was 60%, with parameter values: Radius 1, Neighbors 8, Grad X 8, Grad Y 8 and Threshold 50.

Table 3- Eigenface algorithm test results

Hits %	Confidence
58.97	10.545
58.97	0.185
58.97	2.085
58.97	10.545
58.97	10.545
58.97	6.465
58.97	3.465
58.97	14.645
55.63	16.255
55.63	17.915
55.63	19.715
28.96	38.935
25.63	29.015
58.97	12.775
58.97	12.775
58.97	19.265

IV. RESULTS ANALYSIS

Finally, given the results obtained, it can considered that the algorithm with the best performance in face recognition was Fisherfaces, with a value of 96.6% of correct answers. However, despite having obtained a lower value than Fisherfaces in the number of correct answers, the LBPH with 60% of correct answers did better in the confidence item. The Eigenfaces algorithm with a value of 70% assertiveness obtained the worst result of the three algorithms in the confidence item. Thus, it is up to the user to assess the desired degree of importance and choose between an algorithm with greater assertiveness or greater confidence.

V. CONCLUSION

In conclusion, using the Eigenface, Fisherface, and Local Binary Patterns Histogram (LBPH) algorithms with OpenCV can provide a solid foundation for a facial recognition system. Eigenface is a dimensionality reduction technique that can be used to extract the most important features from facial images, which can then be used to identify faces. Fisherface is another dimensionality reduction technique that can improve the accuracy of the recognition system by taking into account the variation between different classes of faces. LBPH, on the other hand, is a texture-based feature extraction method that captures the local texture of facial images. By combining these three algorithms, a more robust and accurate facial recognition system is developed. However, it is important to note that the success of the system will depend on the quality of

the dataset used for training and testing. Additionally, ethical and legal considerations must be taken into account when developing and deploying such a system.

REFERENCES

[1] Taleb I, Ouis M.E.A, Mammar M.O, "Access control using automated face recognition: Based on the PCA & LDA algorithms", In Proceedings of the 2014 4th International Symposium ISKO-Maghreb: Concepts and Tools for Knowledge Management (ISKO-Maghreb), Algiers, Algeria, 9–10 November 2014, pp. 1–5.

[2] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, "Emotion recognition in human-computer interaction", IEEE Signal Processing Magazine, vol. 18, no. 1, 2001, pp. 32–80.

[3] Choi S, Lee S, Choi S.T, Shin W, "Face Recognition Using Composite Features Based on Discriminant Analysis", IEEE Access 2018, 13663–13670.

[4] C. Lin and K.C. Fan, "Human face detection using geometric triangle relationship", in Proceedings of the 15th International Conference on Pattern Recognition, IEEE, Barcelona, Spain, September 2000, ICPR 2000, pp. 941–944.

[5] Khan M.Z, Harous S, Hassan S.U, Khan M.U.G, Iqbal R, Mumtaz S, "Deep Unified Model for Face Recognition Based on Convolution Neural Network and Edge Computing", IEEE Access 2019, 72622–72633.

[6] K. T. Talele and S. Kadam, "Face detection and geometric face normalization," in Proceedings of the TENCON 2009-2009 IEEE Region 10 Conference, IEEE, Singapore, January 2009, pp. 1–6.

[7] Li L, Mu X, Li S, Peng H, "A Review of Face Recognition Technology", IEEE Access 2020, 139110–139120.

[8] C. Ding and D. Tao, "A comprehensive survey on pose invariant face recognition," ACM Transactions on intelligent systems and technology (TIST), vol. 7, no. 3, 2016.

[9] Ganidisastra A.H.S and Bandung Y, "An Incremental Training on Deep Learning Face Recognition for M-Learning Online Exam Proctoring", In Proceedings of the 2021 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob), Bandung, Indonesia, 8–10 April 2021; pp. 213–219.

[10] Kamenskaya E and Kukharev G, "Some aspects of automated psychological characteristics recognition from the facial image", Methods Appl. Inform. Pol. Acad. Sci. 2021, *2*, 29–37.

[11] G. Tudavekar, S. R. Patil and S. S. Saraf, "Dual-tree complex wavelet transform and super-resolution based video inpainting application to object removal and error concealment," CAAI Transactions on Intelligence Technology, vol. 5, no. 4, 2020 pp. 314–319.

[12] Li Y, Xia R, Huang Q, Xie W, Li X, "Survey of Spatio—Temporal Interest Point Detection Algorithms in Video", IEEE Access 2017, *5*, 10323–10331.