Face Recognition Using Python and OpenCV: Review

Alyaa Abdelbar[*], Hana Haytham[*], Malak Ziad[*], Mariam Amr[*], Maya Haytham[*], Nermean Ashraf[*], Kareem Moussa[*,], M. Saeed Darweesh[*,**]**

[*] School of Engineering and Applied Sciences Nile University, Giza, 12677, Egypt
[**] Wireless Intelligent Networks Center (WINC), Nile University, Giza, 12677, Egypt

DOI: 10.29322/IJSRP.14.01.2024.p14408
https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14508

Paper Received Date: 17th November 2023
Paper Acceptance Date: 24th December 2023
Paper Publication Date: 6th January 2024

Abstract- The human faces are dynamic multidimensional systems that require good recognition processing techniques. Over the past few decades, the interest in automated face recognition has been growing rapidly, including its theories and algorithms. Public security, criminal identification, identity verification for physical and logical access, and intelligent autonomous vehicles are a few examples of concrete applications of automated face recognition that are gaining popularity among industries. Re-search in facerecognition started in the 1960s. Since then, various techniques have been developed and deployed, including local, holistic, and hybrid approaches, which recognize faces using only a few face image features or whole facial features. Yet, robust and efficient face recognition still provides challenges for computer vision and pattern recognition researchers. In this paper, the researchers offered an overview of face recognition, the different used techniques in previous literature and their applications.

Index Terms- Face Recognition, Person identification, Image processing, Survey

I. Background

Facial recognition is a biometric tool. Like otherregularly used biometric technologies such as fingerprint recognition, iris recognition, and finger vein patternrecognition, it identifies a person based on specific physiological features. The introduction of facial recognition in the field of pattern recognition had an impact on the range of applicability, particularly for cyber investigations. This hasbeen possible due to advanced training techniques and progression made in the analysis. Increased demand for a robust security system led the researchers to find work on finding a reliable technology that verifies identities. Facial recognition systems could be the best solution due to theirspeed and convenience over other biometric technologies. The identity of any person is incomplete without facial recognition. Just like any other form of identification, face recognition requires samples to be collected, identified, extracted with necessary information (features), and stored for recognition.

Though the software used varies, the facial recognition process generally follows three main phases. First, the face is captured in real time. The software then determines a number of facial features known as landmarks of nodal points on the face. This includes the depth of the eye sockets, the distance between the eyes, the width of the nose, and the distance from the forehead to the chin. Each software uses different nodal points and can obtain up to 80 different measurements. This data is then converted into a mathematical formula that represents the person’s unique facial signature. Afterward, the facial signature is compared to a dataset if known faces. This can all happen in a matter of seconds [1]. Facial recognition technology has been around since the 1960s. Woodrow Wilson Bledsoe developed the early system that identified photographs by manually entering the coordinates of facial features such as the mouth and nose using an electrical stylus. When given a photograph of a person, the system could extract images from a dataset that most closely resembled it [2].

However, since its inception, facial recognition has been polarizing. Facial recognition technology is widely used in the field of safety and security. It is used by law enforcement agencies to fight crime and locate missing people. Furthermore, face recognition technology is increasingly being used at airport security checkpoints around the world to protect passengers and identify criminals attempting to enter the country [3, 4]. Today, some companies are developing a service using face recognition data platforms to help prevent shoplifting and violent crime. Facial recognition technology is getting faster and more accurate every year. However, its applications do not stop at safety and security. It could also soon be used to make our lives more convenient [3].

Facial recognition is increasingly used in mobile devices and consumer products to authenticate users. College classrooms and testing facilities are using it to take attendance and prevent cheating. Retailers are using it to identify customers. Moreover, some automotive manufacturers are developing ways to use the technology in place of car keys. Facial recognition could also be used for targeting products to specific groups by offering a personalized experience [3]. There are vocal arguments against facial recognition technology, with the biggest being its threat to an individual's privacy. Some cities across the world are already working towards banning real-time facial recognition. That is mainly because facial data can be collected and stored without the person's permission [1].

Yet, the technology is still not perfect as has been demonstrated. It is being widely adopted since it is an advancing technology still seeking to reach high accuracy [5].

A. How Facial Recognition Works

Since computers would not understand faces as humans would, this technology is built on turning face images into numerical expressions, called templates, that can be processed by the computer and then used to compare with other face images. For the matching to be accurate and reap true results, features need to be extracted from an image that makes it unique, so that when compared in the future with other images in the dataset, both images will match only if they share the same features. Formerly, the distances between key points on a template were taken for such processes; however, that was far from accurate [6].

Since computers recognize images as matrices where their numbers represent pixel colors, facial recognition focuses are processing such matrices in a way such that faces can be recognized from the way the numbers are organized. Modern approaches have led to passing the digital face image through a series of "filters" to generate templates that are unique for each face. These filters are used in a form that will produce a distinctive simplified fingerprint for the face being processed [6].

Earlier at the beginning of facial recognition, scientists would pick the filters themselves that would be applied to the images. However, nowadays computers are responsible for that task through deep learning. The filters are selected by giving the system a series of three images, two of each are of the same person and the third is for someone else. Through trial and error, the system must reach the maximum similarity between both images and the least with the third of each triplet in the series. The desired output is a collection of filters that are reliable enough to produce the distinctive face templates needed for facial recognition [6].

B. Facial Recognition Matching Methods

Facial recognition has been approached through various methods over the years. Some of those major methods are feature-based (also known as local), holistic, and hybrid matching [7].

1) Feature-based: In this method, distinct facial features such as mouth, nose, eyes, and more are of importance; thus, they are extracted as key points with their location determined and are geometrically processed on them to be represented later by vectors of distances and angles relating these specific features. Then, pattern recognition techniques are used to match the faces using these measurements [8]. Feature-based matching is considered high-speed since it uses specific features of the face during the process and ex- tracts them before the analysis starts. It also can be modified to be invariant to lighting, orientation, or pose; however, feature detection could be hard which would force the programmer to discard some features that are vital making comparability less accurate [9, 10].

2) Holistic: This approach takes the whole face into consideration rather than focusing on some special features. It deals with 2D face images and works on comparing them directly to each other and correlating their similarities. Eigenfaces is one of the methods used and was developed by Sirovich and Kirby. It is one of the most widely researched face recognition techniques, also known as Karhunen-Loe`ve expansion, Eigen image, Eigenvector, and a principal component [11]. First, a set of images, known as the training set, is inserted into a dataset to be compared later with the created eigen- faces. Subsequently, eigenfaces are generated by extracting features from the faces using the mathematical tool Principal Component Analysis (PCA) [12]. This tool helps reduce dimensionality to finally represent the eigenfaces as vectors of weights. Next, the system receives the unknown image that needs to be recognized and finds its weight to compare with the weights of the training set. For the image to be identified, it must be below a given threshold. Finally, the image in the dataset with the closest weight to that of the unknown one will be the output [8, 9, 13]. Although this method's advantage is that it does not destroy any of the image's information, this makes it computationally expensive since it values all the pixels of the image. Moreover, they do not work best when a face is in very distinct poses or highly illuminated. Still, after multiple enhancements and modifications to make up for its shortcomings, it is considered to work better than the feature-based method [9].

3) Hybrid: This type of matching method employs both feature-based and holistic matching methods on 3D face images [7].

C. Challenges of Face Recognition

Even though face recognition systems have been widely expanded and numerous methods have been applied for face recognition. It still faces many challenges in real-life applications. Here are some of the challenges explained.

1) Posing: People sometimes unconsciously pose differently every time they take a picture. Thus, pose variations cause a serious challenge for face recognition systems which can lead to no or faulty recognition of the face, especially in cases when the dataset has only the frontal view of the face. Face recognition systems can manage only small variations in rotation angles while the pose of the face that is createdby the rotation of the head position or changing the point of view of the camera cause higher rotation angles. As a result, better pose tolerance and the ability to recognize different poses should be taken into consideration in face recognition systems. Techniques aiming for aligning to the image's axis can be used to handle this challenge [14, 15].

https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14508

2) Illumination: Illumination is a challenging condition thatcan heavily affect face recognition systems. As variation in lighting changes the appearance of the face extremely. Even if a person took the same images with the same pose and expression but with varying lighting can appear drastically different. Also, it is shown that the difference between two different images of two different people under the same illumination conditions is lesser than the difference between two images of the exact person under different lighting conditions. Thus, this challenge has attracted the attention of researchers and is widely considered to be hard for algorithms and humans. To overcome this challenge, there are three methods that can be applied, gradient, grey level, and face reflection field estimation techniques [14, 15].

3) Occlusion: Occlusion can be defined as something that blocks the face or part of the face such as a mustache, hands on the face, scarf, sunglasses, or shadows caused by extreme light. Occlusion of less than 50% of the face is referred to as partial occlusion. This challenge degrades the performance of face recognition systems. The holistic approach is one of the methods that can be used for this challenge as it suppressesthe features, traits, and characters while the rest of the face is used as a piece of valuable information [14, 16].

4) Resolution: Varying quality and resolution of the imagesgiven as inputs are considered a crucial challenge. Any image that is below 16 x 16 is considered a low-resolution image, those types of images are used in CCTV cameras in public streets or supermarkets. Also, they do not provide much information as they are mostly lost. As a result, the recognition process goes down drastically. Hence, there is a direct relation between efficiency and the face recognition system as when the resolution increases it gives a better, easier, and more efficientrecognition process [15].

5) Aging: Aging is an uncontrollable process for all human beings. As a result, the Human face is changing over time, it may contain marks or wrinkles which affect the face recognition system. Hence, face Recognition efficiency is waning with age [14]. Having established the context for the study, it is crucial to grasp a clear comprehension of the existing literature in the field. This will aid in comprehending the significance of the study, which will be further talked about in the subsequent section.

D. Contribution

Face recognition systems are widely used in many real-world applications as documented in several research papers which are discussed in detail in the following sections. This paper provides a comprehensive survey of face recognition methods, from traditional feature-based methods to recent deep learning methods and identifies the key challenges that need to be addressed in order to improve the accuracy of face recognition systems. The subsequent sections will provide detailed insights into the various aspects of the system's development and implementation. In the Literature Review section, this paper provides numerous previous research paperson face recognition using different techniques and approaches that will be discussed, which will provide a foundation for the current study. Finally, the conclusion section will summarize the main findings, discuss the limitations of the developed system, and suggest future research directions.

I. LITERATURE REVIEW

A. Brief view of ANN

Before proceeding to understand CNN, which is the most common deep learning algorithm used in face recognition, itis important to understand the architecture of Artificial Neural Networks (ANN) which is the core of the fully connected layerin CNN.

ANN is inspired by the biological brain and its ability to process information. Simply put, its information processing structure is formed through learning where it receivesexamples relevant to a specific application to help improveand adapt by changing the connections between the neurons or the weights they pass through. Thus, ANN is special for its adaptive learning and self-organization. Also, good ANN software can be stable against minor disturbances [17].

The process of ANN is separated into three layers: the input layer, the hidden or middle layer(s), and the output layer. In CNN, the input of ANN is the flattened image or vector. The hidden layers are the actual neurons in the network and perform calculations on the input vector resulting in the output layer. The number of hidden layers and the neurons in themis determined through multiple iterations of forward and backward propagation through the network. Forward propagation or feed-forward network starts from the input till it reaches an output that is compared to a predicted output and backward propagation adjusts the hidden layer neurons' weights depending on the calculated error as shown in Fig. 1 till the actual output equals the predicted one [18]– [20].

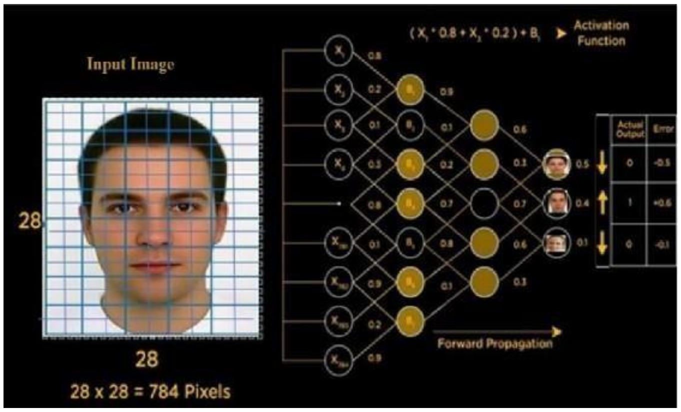


Fig. 1: ANN with error calculation of output for backward propagation [20]

Within each neuron two functions are carried out: the transfer function where the inputs are multiplied by very small non- zero weights and added to another small non-zero number known as

https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14508

bias resulting in a net sum, and the activation function which takes the net sum to produce the neuron's output, or simply put, depending on a certain calculation it decides whether a neuron should be activated or not [21]. There are various types of activation functions: step, sigmoid, tanh, or Rectified Linear Unit (ReLu). Most of the time, ReLU is used as the activation function as it outputs the value when larger than zero and outputs zero when less [22] whose equation is equation 1.

$$f(x) = \max\{0, z\} \qquad (1)$$

Each neuron within every hidden layer carries out these two functions and passes its output to the successive hidden layer if there is one until the output layer is reached. If the final output does not match the predicted one the error is sent back through the neurons to adjust their weights. The output layer has a variable number of neurons but most of them lie under the following classifications: regression in which the output is a single neuron of a continuous number, binary classification where the neuron is either 1 or 0 to signify classes, and multi-class classification which represents various classes [21].

B. *CNN in Face Recognition*

Convolutional Neural Network (CNN) is one of the most used deep learning architectures in face recognition nowadays for its improved performance. It is designed to extract image characteristics. It consists of three main layers: convolution, pooling, and fully connected as shown in Fig. 2 [21, 23].

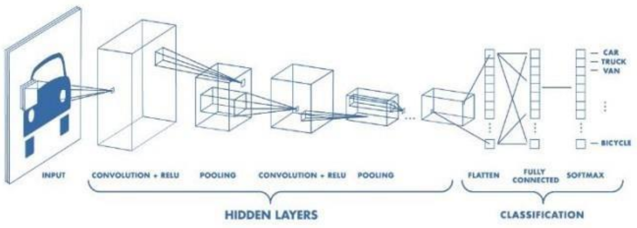


Fig. 2: Convolution Layers [24]

1) Convolution Layer: An image can be viewed as a 2D matrix of its pixels. In face recognition, a filter or kernelis an $n \times n$ matrix extracted, usually sized at (3×3), (4×4), or (5×4). The convolution layer acts as the key building block of CNN, and it is where most of the processing takes place. Convolution is done by having a filter go over the 2D input image and is multiplied by the corresponding $n \times n$ matrix inthe image as shown in Fig. 3. The filter may stride over the input image matrix by 1 pixel as the red box does or 2 pixels as the orange box, then outputs the resulting matrix of convolution known as the feature map. The convolution layer is sometimes followed by one of the activation functions previously summarized before entering the following layer, pooling [18, 23].

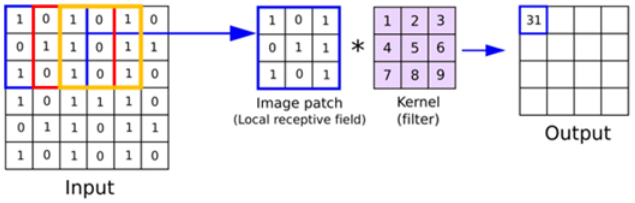


Fig. 3: Convolution Process

If the image is colored then it would be 3D; however, nothing changes since the same process will be done to each of the RGB color matrices and then added together giving a 2D output matrix.

2) Pooling Layer: The pooling layer follows convolution which helps reduce the matrix size and speeds up the processing. Either Max-pooling or average pooling are used as shownin Fig. 4. Max-pooling is done by passing another filter over the feature map and taking the maximum value of the filtered region and outputting it to the final feature map. The filter is usually set to 2×2 with a stride of 2. Average pooling is done by taking the average of the filtered region and is usually applied once before the fully connected layer to reduce the number of learnable parameters. Just like convolution filters, pooling filters may vary in size and strides taken [18, 25].

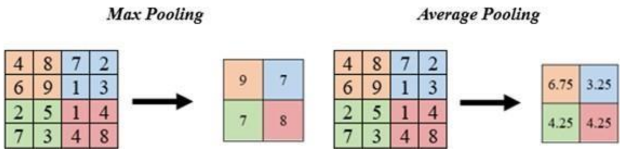


Fig. 4: Pooling types

3) Fully Connected Layer: The max-pooled image passes through the flattening step which unfolds the matrix and turns it into a one-dimensional vector. Passing that final vector ANN's feed-forward network gives the fully connected layer output which may be multiple as shown in Fig. 5 or single output neurons [23].

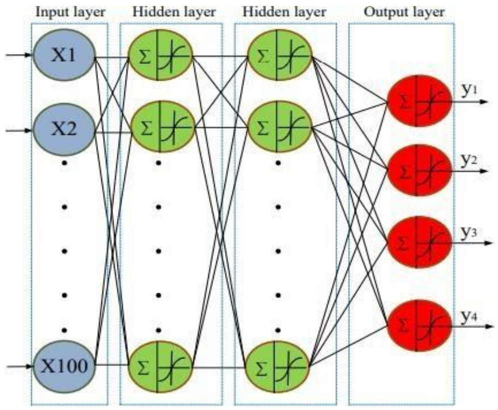


Fig. 5: ANN layers with multiple outputs [26]

CNN layers could be arranged differently depending on the application. Multi-layer perceptron (MLP) is the most commonly used practice when using the CNN algorithm, where multiple convolution and pooling layers are used interchangeably for feature extraction then finally followed by fully connected layers. The output of a certain CNN structure is trained first and then implemented if high accuracy was acquired [27].

CNN was the algorithm used in paper [28], where a variation of the earlier explained layers was organized to reach a high accuracy of face recognition. The proposed design used the ORL dataset as input data and resized them into 32×32 pixel-sized images and passed through 6 layers of feature extraction and processing. The layers were organized as follows: first convolution layer (16 feature maps and 3×3 kernel dimension) in addition to the ReLU activation function, MaxPooling layer (2×2 kernel dimension), second convolution layer with the same kernel size but doubled feature maps count, another pooling layer identical to the previous one; finally, two fully-connected layers followed each other (the first had 3000 neurons while the second had 40). This paper's design divided the dataset into 4 classes (A to D) and increased the percentage of training data used to test the difference in accuracy. The results showed that by using 10% of the dataset's class A for training, the accuracy was 93.9%. Class B used 30% of data for training and got 95.7% accuracy, class C used 50% leading to 97.5% accuracy, and class D used 50% with 98.3% accuracy. Although the accuracy of the overall design was better than other designs and the accuracy kept increasing by increasing the training data size, the computation load was very high and memory usage was significant.

Another paper used CNN for the designing of a facial recognition system for grey-scale images in paper [29]. The researchers used a variation of CNN's stack of layers that work independently and ordered them in a way that ensures high accuracy as well as efficiency. The used network consisted of two convolution layers (CONV), to process the input data, each followed by an activation layer (ReLu) that decides whether the neuron's input will be significant during the process of prediction; and a max pooling layer that helps reduce the size of the intermediate image usually using subsampling. Following each CONV layer, two regularization techniques (batch normalization: which speeds up the training process by using higher learning rates, and dropout: which eliminates weakly connected neurons) were added as well in the training process to enhance performance. The dropout technique was added again after the fully connected layer, which combines learned features, to avoid overfitting. Finally, the output layer uses the SoftMax function to calculate the probability of each class, which all needs to add up to one. In that paper, the proposed CNN had a fixed image size of 32×32×1, where 1 referred to gray-scale coloration. The program was made in Python using the Tensorflow Deep Learning framework and the OpenCV library. The dataset used was ORL's [30] which captured a total of 400 frontal view faces, between 1992 and 1994, classified into 40 distinct classes of 10 images per class. Their size was 92x110 grey-scaled pixels. The dataset was divided so that the training set used 6 images per class and the other 4 were used for testing. The proposed CNN validation was tested using the Categorical Crossentropy loss function. After being applied to the dataset, the accuracy reached 99.78% for training and 98.7% for validation accuracy which is considered high to that of Eigenface and ICA algorithms [29]. The proposed CNN algorithm in that paper could be trained and tested on colored images and redesigned if extra layers are needed to reach high accuracy, as further research. Another research paper [31] used CNN's various structures throughout the whole algorithm designed for face recognition and registration of learners during online classes to ensure their authenticity. The proposed architecture of this paper, IndepthNet-19, used multi-task cascaded CNN to detect the face since it worked faster than Viola-Jones and had fewer restrictions due to the environment of the image. For the process of feature extraction deep learning CNN was used for its simple strategy. Then the identification technique compared the face detected with that in the dataset and finally SoftMax was used to represent the probability function for all classes.

The first four datasets used for comparing accuracy varied in the number of classes or subjects, image sizes, face expressions, and environments. The datasets were Faces94 (1h480×200 pixels, 152 subjects, minor expression changes), Faces95 (196×196 pixels, 72 subjects, some expression changes), Faces96 (196×196 pixels, 151 subjects, some expression changes), and Grimace (180×200 pixels, 18 subjects, major expression changes), adding up to 393 faces in the dataset with 20 images per subject or class. The accuracy achieved with this algorithm, compared to ResNet-50, DenseNet-201, VGG-16, and FaceNet, for each of the datasets were as follows: 100% for Faces94, 99.86% for Faces95, 99.54% for Faces96, and 100% for Grimace [31].

For distance learning and registration, a different dataset was acquired from the live detection of faces in online classes. This time, another CNN model was introduced that solves the issue of barely having enough datasets by using data transformation and augmentation to generate images with different intensities, brightness, etc. The feature extraction was handled usinga multi-descriptor and the probability was measured using SoftMax and support vector machine (SVM). Additionally, the cosine similarity function was used to measure the performance of the model. Two tests were applied to this dataset, the first was where the faces were in frontal view, with different expressions and slight rotation, which reaped an accuracy of 100%. The second test used frontal faces as well; however, the rotation was larger with major expression changes, different backgrounds, and a large head scale, giving an accuracy of 88.88% [31].

C. *Eigenfaces and Principal Component*

Eigenface is one of the most widely researched face recognition techniques. It is also known as Karhunen-Loe`ve

https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14508

expansion, Eigen image, Eigenvector, and a principal component. The Eigenfaces method is based on the principal component analysis (PCA) [32, 33]. Sirovich and Kirby employed Eigen- faces and PCA to efficiently represent face images [34].

The principal component analysis is a statistical dimensionality reduction method that provides the optimal minimum- square linear decomposition of a training set [35, 36]. It is a widely used method for detecting patterns in large- scale images and is utilized in image compression and face recognition for two major reasons [37], [39]. First, it has a low noise sensitivity. Second, because it operates in a smaller dimensional space, it has high efficiency. PCA technique is applied to reduce data dimension and expose the most effective low-dimensional facial pattern structure. Hence, eliminating unnecessary information and decomposing the face structure into uncorrelated components known as Eigenfaces. After- ward, Face images are stored in a 1D array representing the weighted sum of the Eigenfaces (feature vector). The main advantage of this approach is that the information needed to identify the person is reduced to 1/1000th of the existing data. However, a full front view of the face is required; otherwise, the recognition output would be incorrect [40].

The Eigenfaces method strategy, on the other hand, includes extracting the face's distinct characteristics and expressing the face image as a linear combination called ”eigenfaces” obtained from the feature extraction process [41]. The face image key features are then calculated. After that, the test and trainingimages are matched using Euclidean distance. If the Euclideandistance is short enough, the person is recognized; otherwise, the person is unidentified, and his picture is assumed to belong to another person whose image is yet to be known [40].

Mu¨ge Arkc¸ and Figen Zen used the MATLAB program to build a facial recognition system based on the Eigenfaces method as shown in Fig. 6.

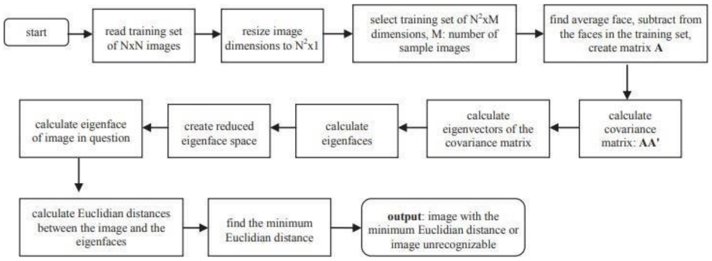

start
read training set of NxN images
resize image dimensions to N²x1
select training set of N²xM dimensions, M: number of sample images
find average face, subtract from the faces in the training set, create matrix A
calculate covariance matrix: AA'
calculate eigenvectors of the covariance matrix
calculate eigenfaces
create reduced eigenface space
calculate eigenface of image in question
calculate Euclidian distances between the image and the eigenfaces
find the minimum Euclidian distance
output: image with the minimum Euclidian distance or image unrecognizable

Fig. 6: Flowchart of the Eigenfaces algorithm [40]

The study included a dataset of 3040 images, which included twenty images of one hundred fifty-two people with diverse facial expressions, backgrounds, and lighting conditions. They had challenges with some features such as a mustache, beard, eyeglasses, and the image background. However, their face recognition system was successful, with a success rate of 94.74%. If more information is used, their success rate might increase. For instance, the face triangle [40].

D.Deep Learning-Based Representation for Face Recognition

A variety of issues are being resolved using an effective approach which is deep learning. Although it is still in an early stage, it is fast expanding and transforming worldwide. Deep learning is a subset of machine learning, which uses a set of algorithms to try to simulate high-level abstractions in data by using multiple processing layers with complex structures as it gives a good estimation of complex functions [42, 43]. The authors of [42] created a face recognition system that initially detects and recognizes the input by setting up a camera with the software used to capture the images; then, provides the input to be detected. Images, recorded videos, and real-time videos can all be used as input. Following that, a classifier has been trained. A CNN classifier is used for face recognitionand to recognize the input data. The next step is to run the different types of recognition by a different set of Python scripts. Finally, to implement the process of recognition, the Python script will import from the previous step the classifier to conduct the recognition for the person from the camera or from an image. Also, for face detection, to find and locate faces in an image or video. Haar feature-based cascade classifiers are used specifically, the Haar Cascade classifier for the frontal face. Haar cascade is a classifier that serves just to identifythe object for which it has been trained from the source. It is carried out by overlaying the positive image (image of the target object) on a set of negative images (images that do not contain the object) [44]. The paper presents four types of datasets for face recognition including distance face detection, lighting condition, and accuracy of face recognition based on image and real-time video. This model is trained using a large number of images per candidate. Also, the use of the CNN approach led to a huge dataset and enhanced overall accuracy. The dataset used was via recognition of three people multiple times; Multiple photos were uploaded into the system.To undergo recognition, the person should have appeared in those photos 20 times, and for the video, people will show themselves to the webcam at different locations. Results foundthat the face can be recognized when the distance is lengthenedto more than 60 cm. If it is less than 60 cm the system will hardly detect the face. Besides, When the light intensity is minimal, the accuracy to detect the face is considerably low in comparison to higher light intensity. The overall accuracyof the system for face recognition from images is 91.7% and for the real-time video 86.7%.

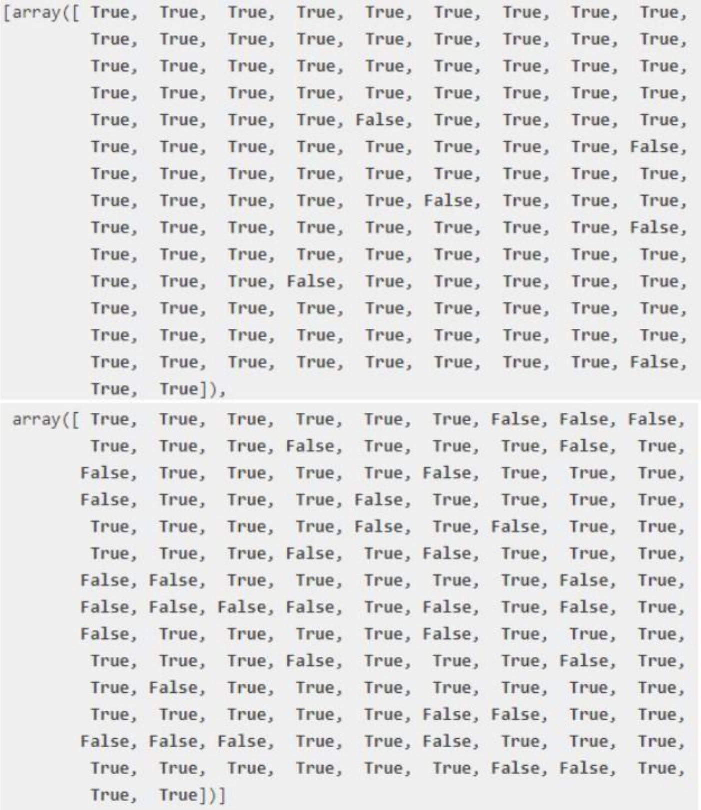
Since deep learning has been taking over the field of biometrics day by day, it has offered a quality answer in terms of recognition performance. The goal of [45] is to study deep learning-based face representation under a variety of circumstances, such as lower and upper face occlusions, misalignment, different angles of head poses, changing illuminations, and flawed facial feature localization. Consequently, to achieve this, two different widely used deep learning models, Lightened CNN and VGG-Face, have been taken into consideration in this paper for the extraction of face representation. Those two models illustrated an important remark that the deep learning model can tolerate many types of misalignment and localization errors. In this paper, data is entered into a convolve filter in different multiple levelsthat accordingly detect underlying high levels of representation from data that has been labeled or not. The tested conditions are rare in the training datasets of CNN models as a result the performance may be affected. There are two approaches defined in this paper for face identification. First is VGG-FaceNetwork. In this model, the training is done on 2.6 million images of the face which are obtained from the web of 2522 people. Besides, there are 16 convolutional network layers that make up the network, three fully connected layers, five max-pooling layers, and the final linear layer having Softmax activation. Color image covers of size 224×224 pixels are taken as input by VGG-Face then regularization is used in the completely connected layers. The second is Lightened CNN. Lightened Convolutional Neural Network is familiar with dual distinct model types. The AlexNet is model (A), comprising four distinct convolution layers and 3962K factors using the four max-pooling layers, Max-Feature-Map (MFM) activation functions, two fully connected layers, and a linear layer with Softmax activation in the output. The second network (B) is encouraged by the Network in Network model, and it involves 3245K factors comprising five distinct convolution network layers that use MFM activation functions, five max-pooling layers, four convolutional layers for reducing dimensionality, linear layer includes activation with the Softmax having two fully-connected layers in the output. To undergo the segment of tests and results that illustrate the datasets and experimental systems, Using theAR Face dataset, 126 people contributed with frontal facephotos with a resolution of 764×574 pixels and a rangeof expressions, illuminations, and occlusions. Each subject participated in two sessions separated by two weeks. Taking into account that there are no restrictions such as a scarf,make-up, headwear, hairstyle, etc. A single image of each individual from the first session with a normal expression is utilized for training. As mentioned, the crucial objective of this test is to assess the robustness of deep CNN-based features against occlusion. To achieve this, in each testing session, two images are used for each subject for testing, making the posture constant to test how an upper face occlusion having sunglasses will affect and in contrast another one wearing a scarf to test the lower face occlusion impact. To generate the output of related image transforms smeared on trained models, the mean picture produced after being subtracted from the VGG-Face training set was used. To verify the purpose of the paper, the AR dataset is complemented by five experiments. The first two experiments involve training and testing during the first session, while the remaining experiments train using samples from the first session and test using photos from the second session.

E. OpenCV

OpenCV was originally developed by Intel in 1999 and itis the abbreviation for Open-Source Computer Vision Library, a software library for computer vision and machine learning. It offers many tools and features that let developers develop apps with advanced computer vision capabilities and are also designed to be highly efficient and optimized for real-time applications. It offers interfaces for many computer languages,including Python, Java, and MATLAB/Octave, and is written in the C++ programming language. The library provides more than 2500 optimized algorithms that may be used to complete tasks including processing images and videos, detecting, and tracking objects, extracting features, camera calibration,3Dreconstruction, machine learning, and more. The ability ofOpenCV to work with several types of image and video datais one of its key features Supports a wide variety of image formats, including JPEG, PNG, TIFF, BMP, and many others, as well as provides functions for capturing video from camerasor reading video files [46, 47].

1) OpenCV Applications: According to paper [48], During the COVID-19 pandemic people wore face masks which makes it hard for AI to detect faces without using common facial features such as nose, eyes, mouth, chin, etc. Henceits main objective was to show specific techniques to detect and recognize faces with and without masks to have accurate face recognition systems work efficiently even in an extreme scenario such as the COVID-19 pandemic [48].

Any process of Image Learning and recognition needs libraries to help the process. The libraries used by paper [48] are face recognition which is a component used in their program built using Dlib's facial recognition with deep learning which has a high accuracy. Dlib is a toolkit created in C++ that includes many machine-learning algorithms and tools for developing sophisticated software to solve real-world problems. Another key module and library used in their program is PIL, which stands for Python Image Library. It supports a wide range of image formats, including JPG, PPM, and PNG. PIL module and face recognition library available for Python were chosen as they face fewer challenges compared to other methods as those methods cannot operate for images of people wearing a mask as they require the whole face to *appear.face recognition* module is extensively used to recognize and identify faces from a given image in Python and works with three steps. In the first one, by using the load *image file()* method the needed image is imported and then converted into NumPy array by the *face recognition* library. Landmark algorithm called face landmark estimation is used to search and find the features found in a specific image. The algorithm works in multiple ways, the one used by the programs is by calculating and storing one hundred and twenty-eight special points of the face ranging between the top of the chin to the sides of the eyes besides covering the eyebrows then storing these one hundred and twenty-eight values in an enormous array and they are understood by the algorithm as RGB values. Once the model has been trained with several images of the same face taken from various perspectives, it becomes proficient at identifying that particular person. The second step is the *face encoding's* function. After importing the image, it will be checked if known face encodings match with any face encodings by cross-checking those two arrays and the previous one hundred twenty-eight components mentioned. This comparison is done between the test and reference images using the compare faces function of face *recognition library* function, then an array is received after the comparison is done as shown in Fig. 7. These test images contain people who wear a mask. When the faces are compared afterward if they are within the tolerance limits which vary between the different recognized samples, an accurate result is produced [48].



```
[array([ True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True, False,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True, False,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True, False,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True, False,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True, False,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True,  True,  True,  True, False,
         True,  True]),
 array([ True,  True,  True,  True,  True,  True, False, False, False,
         True,  True,  True, False,  True,  True,  True, False,  True,
        False,  True,  True,  True,  True, False,  True,  True,  True,
        False,  True,  True,  True, False,  True,  True,  True,  True,
         True,  True,  True,  True, False,  True, False,  True,  True,
         True,  True,  True, False,  True, False,  True,  True,  True,
        False, False,  True,  True,  True,  True,  True, False,  True,
        False, False, False, False,  True, False,  True, False,  True,
        False,  True,  True,  True,  True, False,  True,  True,  True,
         True,  True,  True, False,  True,  True,  True, False,  True,
         True, False,  True,  True,  True,  True,  True,  True,  True,
         True,  True,  True,  True,  True, False, False,  True,  True,
        False, False, False,  True,  True, False,  True,  True,  True,
         True,  True,  True,  True,  True,  True, False, False,  True,
         True,  True])]
```

Fig. 7: The received array after comparing the reference and test images [48]

The final step is using ImageDraw and *drawr˙ectangles* function, when a face is recognized, these functions draw a box around that face. Additionally, the use of the face *recognition.face landmarks()* function pinpoints the precise locations of particular facial features [48].

Their results were accurate with 70% to 85% and takes from 20 to 30 seconds to identify a face. The challenges faced by this program where some images could not be recognized due to lighting conditions and pixelated images. The model only works on a predefined reference dataset. Hence when a faceis detected but not available on a given dataset it will show Unknown as an output. It is mentioned that in the future work can be done to overcome these scenarios and would work on implementing emotion recognition while wearing masks [48]. Another research paper [49] used OpenCV along with Tensorflow to detect whether people wear facemasks or not. TensorFlow was developed by Google and released in 2015. Itis an open-source machine learning framework used in creatingand training neural networks [50]. TensorFlow has a collection of tools, libraries, and community resources to ease the building of deployment of machine learning-powered applications. For the detection of face masks on people, this paper used OpenCV, TensorFlow, Caffe-based face detector, Keras, and MobilenetV2.By using the Caffe deep learning framework, a Caffe-based face detector which is a face detection system was built [49]. Caffe is maintained and developed by the Berkeley Vision and Learning Centre (BVLC) with the assistance of an active community of contributors on GitHub [51]. It utilizesthe strength of deep neural networks to precisely locate and recognize human faces in images or videos. Keras is an open-source library that provides for artificial neural networks a Python interface. Keras serves as an interface for the libraryof TensorFlow [52]. MobileNetV2 is a convolutional neural network (CNN) architecture created by Google researchers as an upgrade to the original MobileNet model. It is particularly made for effective and precise image categorization on embedded and mobile devices with constrained computational power. The method followed in this paper was firstly collecting the dataset from the Kaggle Repository, analyzing them then splitting them into training and testing data, training a model to detect face masks by using the default OpenCV module for acquiring faces then training a Keras model to detect face masks. OpenCV was used in obtaining the face and was trained to identify the names of the people not wearing the mask by referring to the dataset. Lastly, send an e-mail to the person not wearing a mask using smtplib. Their dataset contains 3835 images,1916 of them have people with masks, and the rest of the images of people without masks on them [49]. Kaggle is an internet-based platform for data scientists and machine learning enthusiasts. Users of Kaggle can work together, access and share datasets, use notebooks with GPU integration and compete with other data scientists to solve data science problems [53]. With the help of the strong tools and resources it offers, this online platform was established in 2010 by Anthony Goldbloom and Jeremy Howard and bought by Google in 2017 [54]. smtplib is a module provided by Python which defines SMTP as the abbreviation for Simple Mail Transfer Protocol. Sending e-mail and routing e-mail between mail servers is handled by this protocol.

Paper [49] generated several models, the first was a base model using Keras and MobileNetV2.Also, a head model is composed of a network with 128 layers, ”Relu” as theactivation function, and a dropout of 0.5, followed by a network with 2 layers and ”softmax” as the activation functionthat is generated on top of the base model. By combiningthe three layers the training models are produced. Then the generated model is trained with the labeled dataset whichis split into 75% images of it used for training and 25%percent for evaluating the model accuracy, then after training the model, it is used for identifying face masks on human faces. The trained model is loaded, and input is given to itin the form of images of people with or without masks oras a continuous video stream. The frame of the video or the images given is sent to the default face detector module forthe detection of the human face, resizing occurs to the image or video frame. Moreover, the output is given with only the face cropped without the background. This face is given to themodel that is trained as an input and then outputs whether there is a mask or not. Lastly, there is a model that is trained with the human face using OpenCV. The images used for training the model are provided with the name and email address of that person. The process of detection is done as follows when an input image is given to the CV model, it identifies the face of that person and then asks the user to provide the name and email address of that person in which this data will be stored in the dataset. The face will be compared with the person present in the dataset, if it is recognized and matched, a bounding box will be drawn over his face with his name and email on it. If the mask is worn, the word “Mask” will appear below the bounding box, and “No Mask” will appear for those not wearing a mask. Also, If the person’s face is stored in the dataset, it detects the name of who is not wearing the mask and an email will be sent to that person notifying them that that they did not wear the mask. The paper suggests that this system can be applied in public places, for instance railway stations, malls, and places with a great number of workers [49].

2) LBP and LBPH algorithm in OpenCV: It was mentioned earlier that OpenCV library, Currently, the library has more than 2500 optimized algorithms [47]. One of the algorithms is HOG which is mentioned in this paper. Additionally, there are also other algorithms such as Local Binary Patterns (LBP) and Local Binary Pattern Histogram (LBPH) algorithms that are used in a variety of applications, such as face and object recognition, and can extract features from images. There are various research papers that mentioned these algorithms in their face recognition systems or application.

Local Binary Patterns (LBP) is a texture operator that is derived from a generic definition of texture in a local neighborhood and the original LBP operator was introduced by Ojala et al. [55]. It is defined as a grayscale unvarying texture measure. LBP texture operator has been a prominent method in many applications as a result of its discriminatory power and computational simplicity. The computational simplicity feature facilitates the analysis of images in difficult real-time conditions. Its crucial feature is its resistivity to monotonic grayscale change, for instance, changes in lighting. Local Neighborhood refers to a particular region or area surrounding a pixel in an image and is defined by choosing a crucial region around each pixel in the image. This region’s size is determined by the radius parameter, which specifies the distance between the center pixel and its neighbors [56, 57]. The original LBP operator generates labels for picture pixels by thresholding the 3×3 neighborhood of each pixel with the center value and then converting the result to a binary integer. The histogram of these various labels can then be used as a texture descriptor. Before performing the LBP operation, it is required to train the algorithm with a dataset of facial images of the person that needs to be recognized recognize, as well as to assign an ID to each image in order for the algorithm to utilize that information in the output result [58, 59].

Fig. 8 shows The LBP operator selects a part of the image that is 3×3 pixels in size, which can also be modeled as a 3×3 matrix having the intensity of each pixel (0-255). The central value of the matrix is utilized as a threshold to define the new values from the 8 neighbors. A new binary value is set foreach neighbor of the threshold .0 if the value falls below the threshold and 1 if it exceeds the threshold. Consequently, the matrix will only contain binary values that, when concatenated, generate a new binary value. The new binary value is then converted to a decimal number and set to the matrix’s central value, which is actually a pixel from the original image [59, 60].

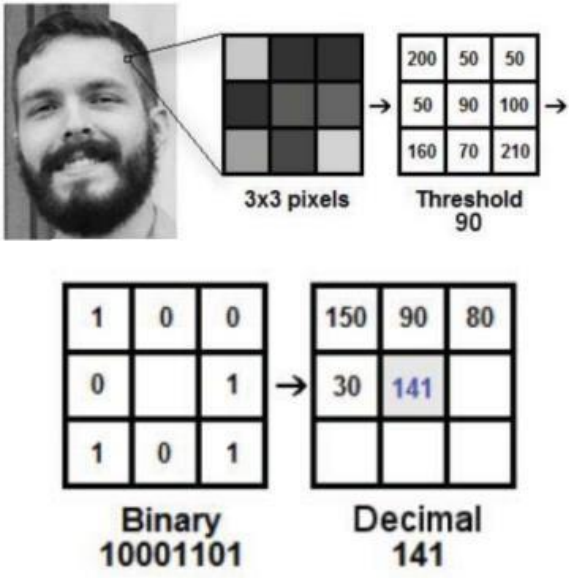


Fig. 8: LBP Operator Example [59]

The local Binary Patter Histogram (LBPH) algorithm is a face recognition method that extracts features from faces using the LBP algorithm. In a training set, the LBPH method generates a histogram of LBP values for each face. Then, using this histogram, faces in new images can be recognized. A histogram is a statistical approach that counts the number of times each color appears in each square [59].

The algorithm makes use of the following four parameters: radius, neighbours, and grids X and Y. The radius, which indicates the radius around the center pixel and is typically set to 1, is used to construct the circular local binary pattern. Neighbors is the number of sample points used to construct the circular local binary pattern; it is typically set at 8. Grid X determines the number of horizontal cells; it is typically set to 8. Grid Y indicates the number of cells that run vertically; it usually is set to 8. To extract the histograms for face recognition, the image is separated into numerous grids with grid X and grid Y parameters. Each histogram from each grid will have 256 positions reflecting the occurrences of each pixel intensity. The final histogram is formed by concatenating each histogram and representing the characteristics of the original image. Each final histogram is utilized to compare each image from the training dataset to the final histogram of the given input image when performing face recognition. The algorithm returns the image

https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14508

with the closest histogram after comparing two histograms from the input image and the training image. Fig. 9 represents an example of extracting histograms [59].

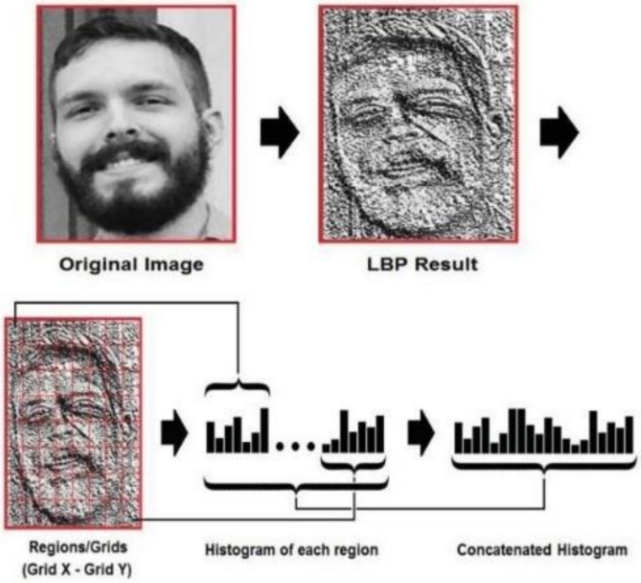


Fig. 9: Extracting Histograms example [59]

F. Face recognition using HOGs algorithm

Histogram of the oriented gradient is one of the best descriptors used for shape and edge description. HOGs are image descriptors invariant to 2D rotation, occlusion, and extreme lightning conditions. Other deep learning techniques use training classifiers to distinguish people. Moreover, HOG assumes that even without perfect knowledge of the corresponding gradient or edge locations, the distribution of local intensity gradients may frequently represent the appearance and shape of local objects quite effectively. It can be applied by segmenting the image window into smaller spatial areasand building a local histogram of gradient axes across each area's pixels [61]. HOG is one of the feature extractors usedin computer vision and image processing. It is described as the distribution of edge directions in an image. The image is then divided into small cells. Several pixels are present in every cell accordingly a histogram of gradients is formed histogram of the gradient is made. The HOG is contrast normalized for greater accuracy by estimating intensity over a bigger area of several cells known as a block, then this value is usedto normalize all cells within that block. The result that has been normalized performs better when light and intensity are variable. HOG descriptors are superior to other descriptorsin that, with the exception of object orientation, they are invariant to geometric and photometric modifications. It isespecially well suited for image-based people detection [62]. Applications for histograms of oriented gradients (HOG) can be found in the fields of object and pattern recognition since they can extract important information even from the images that are obtained under garbled environments. It is therefore well suited for tackling the facial recognition problem. The feature extraction process of HOG is based on extracting information about the edges in local regions of a target image [63, 64].

This system encodes the picture using the HOG algorithm to create a simplified version of the image. After the isolation of faces in an image, utilize a method called the face landmark estimation algorithm developed by Vahid Kazemi and Josephine Sullivan to reduce the chance that faces with varied turns do not belong to the same person [48]. The basic idea is to identify 68 specific points on the image. However, a machine learning system is trained to find those 68 specific spots on any image. No matter which way the faces are turned after using this algorithm, the eyes and mouth can always be centered. In addition, direct comparison between unknown faces and known faces that have already been trained and stored in a dataset is the simplest method for face recognition. 128 metrics for a face are produced using a Deep Convolutions Neural Network that has been trained. Therefore, all that is needed to obtain 128 measures for each face is to send the face photos via the network. In order to improve the accuracy of these 128 measurements and make it simpler for us to match them with unidentified faces, any number of practice photographs can be run with different lighting setups. Then train a classifier that can take in the measurements from a new test image. Eventually, the classifier result will be the name of the matched person. The process of this technique is done by sharing the whole face image into a cell (small region or area). A histogram of pixel edge direction or direction gradients is generated for each cell; finally, the histograms of the whole cells are combined to extract the feature of the face image. The mathematical description of the HOG method: Let G_x andG_y be the horizontal and vertical components of the gradients, respectively. These are computed by using intensities of the pixels $I(x,y)$ at positions (x,y) as in the following equations:

$$G_x = I(x+1,y) - I(x-1,y) \qquad (2)$$

$$G_y = I(x,y+1) - I(x,y-1) \qquad (3)$$

The magnitude and angle of the gradients are computed as:

$$M_{(x,y)} = \sqrt{Gx^2 + Gy^2} \qquad (4)$$

and

$$\theta = \frac{G_y}{G_x} \qquad (5)$$

Where $M_{(x,y)}$ is the magnitude of gradients, and θ is the angle of the gradient at the given location. Tri-linear interpolation divides the gradient's strength and each cell's pixel's direction into nine bins. To extract the feature of the face image, the histograms of the entire cells are combined after each cell's histogram is generated pixel by pixel using direction gradients. A combination of several histograms of oriented gradients (HOG) was suggested by Karaaba et al. in their study [61, 65] to carry out a reliable face recognition system, "Multi-HOG" is the name of this method.

https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14508

There is a new technology that can be used to accelerate the overall speed of object detection using the HOG descriptor. Two main modifications are the introduction of a lookup tablefor computing the oriented gradients of an image. Another is the application of integral images for HOG feature calculation. An integral Image is also known as a summed area table, which is commonly used for quickly and efficiently generating the sum of values in a rectangular subset of a grid, it allowsfor very fast feature evaluation and image input for object detection [66, 67]. Assumed the 2D grid data is of size $W \times H$, its value at location (x, y) is $i(x, y)$. Its integral image value at (x, y) is denoted as $ii(x, y)$ then it can be calculated using equation 6.

$$ii_{(x,y)} = \sum_{x' \leq x, y' \leq y} i_{(x',y')} \qquad (6)$$

An example of doing an integral over asmall area is shown in Fig. 10 [68].

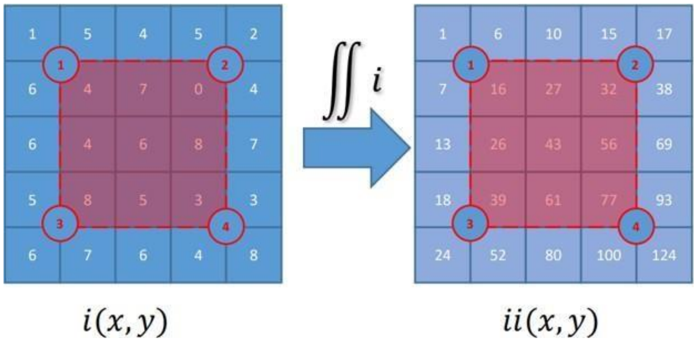


Fig. 10: Calculation of an integral image for a 5×5 grid region. The shield subregion is the area of interest [68]

To calculate the area, sum V_R of an interested region R (shaded area in Fig. 10, in this paper do not have to add upthe value of every single point inside that area. Instead, its integral image representation can be used to compute the sum easily as shown in the following equation [68].

$$V_R = ii(4) + ii(1) - ii(2) - ii(3) \qquad (7)$$

Equation 7 is an oriented gradient map of an image that comprises a 2D grid of bin numbers (for zero-based, these numbers are from 0 to Nb-1 m where Nb is the number of angular bins. A 2D grid of bin numbers makes up an image oriented gradient map. The HOG features for a region inside this map can be constructed using this map. The HOG feature in a region should be built in order to ascertain whether it contains the object of interest. Fig. 11 illustrates how a region’sentire feature representation is often created by joining the oriented gradient histograms of its overlapped sub-areas.

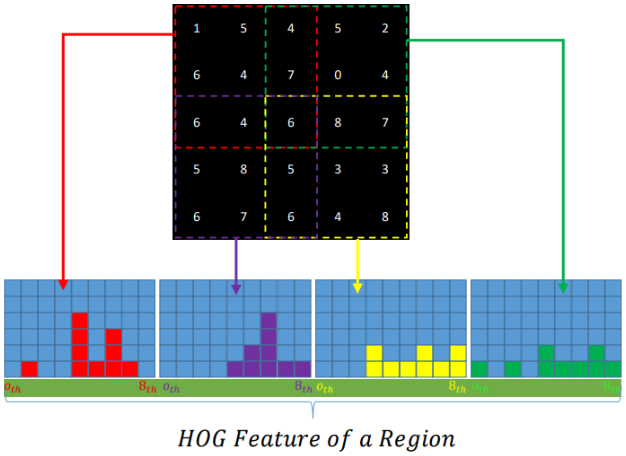


Fig. 11: The HOG features of an interested region are generated by connecting the HOG features of its four sub-areas. The four histograms are the features of the four overlapping areas as indicated by dashed rectangles [68]

By using the integral image approach to calculate the HOG features, one can create integral images for HOG. First, though, it is important to remember how the histogram for each area inside a region is calculated. The number of bins with the same values inside a given area is counted and grouped to create a histogram, which has a length equal to the number of bins. In the first histogram of Fig. 8, certain bins are empty indicating they have zero value. Examples of these bins include (0,2,3, and 8 bins).

SIFT (Scale Invariant Feature Transform) has emerged as a cutting-edge technology for extracting distinctive features from images, to be used in algorithms for tasks like matching different views of an object or scene. Moreover, it achieves variance to scale changes by extracting key points at the local extreme of the scale-space representation of the image, then each key point is represented using histograms of image gradients, in the sequel HOG descriptor. HOG descriptors have also been proposed for pedestrian detection [68, 69]. In these approaches, objects are assumed to be at a fixed scale and are divided into small, connected regions at fixed positions. Then, for each region, a HOG descriptor is obtained, and the combination of these descriptors is used to represent the object. SIFT has also been recently proposed for face recognition [70] however this approach totally differs from ours. In Bicego’s algorithm, key points are located at the local extreme of the scale space as in the original Lowe’s approach [71]. The main problem of this approach is that there is no control over the number, position, and scale of the key points. However, in our algorithm, the key points represent specific landmarks that are detected first as explained below.

Once facial landmarks are detected, HOG descriptors are used to represent them. The first step is used by SIFT to achieve invariance to scale changes. This is done by extracting SIFT features only at the local extrema of the scale-space representation of the image. The next step aims to obtain image rotation invariance. To that end, at each extremum of the scale-space

https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14508

representation, SIFT finds the dominant direction. While these two techniques have proved to be very useful for images that are arbitrarily scaled or rotated, the fact is that these normalization stages remove information that might be useful for recognition when images are not scaled or rotated. In this paper, it was assumed that the exact location of both eyes is known prior. To detect the eyes precisely, an algorithm was developed that uses a mixed approach of boosted classifiers and again HOG descriptors [69, 72]. However, this problem can be deemed as precise face localization, and it is not treated here. Since the exact location of the eyes is used to normalize faces, no changes were expected in either scale or rotation. For this reason, the two first steps of the SIFT algorithm are skipped and only the last step from Lowe's approach, the keypoint descriptor, is adopted [71, 73].

G. Design and implementation of face recognition system in MATLAB using the Features of Lips.

A system that uses faces to identify people is known as a face recognition system. A quick face identification technique with reliable results is described in this study. One of the bio-metric systems that can be used as the basis for a real system is lip tracking. In addition, lip tracking has the benefit of making the system safe since a person's utterances are distinctive and challenging to imitate.

The figure known under 'Equal error rate (EER)' is made to find performance analysis of the open-set speaker identification system. Moreover, for the speaker identification task, word-level continuous-density HMM (Hidden Markov Models) structures are built. They also modeled it usinga different HMM by making each speaker in the dataset represented by the feature sequence taken from the lip stream while saying a secret phrase. In this report, speakers' visual utterances were pre-recorded and saved in a dataset for later verification. Finally, utilizing hybrid edges, it is possible to identify the mouth region and important spots. PIPES AND FILTER architectural style was employed to construct thelip motion features extraction for speaker identification. This paper was divided into four stages: the first stage involves extracting the face region from the original image; the second stage involves extracting the mouth region by removing the background: the third stage involves extracting the key pointsby using the centralization of the lip as the origin of coordinates, and the fourth stage involves storing the obtained feature vector in the dataset. Adding to that, the outside lip border is fitted with five points.

Moreover, they tested their technique using sequences of various speakers. They were collected under normal, irregular lightning circumstances and were unaltered. The sequence's images are RGB (8 bits/color/pixel) and include the area of the face that extends from the chin to the nostrils. Additionally, they consider that light originates from above and that the head can rotate as long as the mouth's openings are visible. The implementation section explains critical processes for creating and implementing an underwriting system. It contains three phases. Firstly, video input is converted to frames during the face region recognitionphase, and the faces are identified along with skin color blocks,noise reduction, illumination adjustment, color transformation,and height-to-width ratio detection. In the first step in this phase, skin-color detection is used in a real-time system for face recognition, together with color space transformation and luminance-based illumination adjustment. computes the input image's average brightness Y_{aveg} to modify the distribution of skin color.

$$Y_{aveg} = \sum Y_{ij} \tag{8}$$

Where

$$Y_{ij} = 0.3R + 0.6G + 0.1B$$

i and j are the pixel indices, and Y_{ij} is normalized to the range of (0, 255). The adjusted picture C_{ij} can be obtained using the following equations to correspond with Y_{aveg}.

$$Rij'=Rij.t \tag{9}$$

$$G'_{ij} = (G_{ij}).\tau \tag{10}$$

$$C_{ij} = \{R'_{ij}, G'_{ij}, B'_{ij}\} \tag{11}$$

Where

$$t = \begin{cases} 1.4, Y_{aveg} < 64 \\ 0.6, Y_{aveg} > 192 \\ 1, otherwise \end{cases} \tag{12}$$

To simplify computation, only the colors of R and G are compensated. Chrominance C_r can accurately represent human skin; thus, in order to simplify computation, just take C_r intoaccount while performing a color space change. C_r is definedas follows:

$$C_r = 0.5R' - 0.419G' - 0.081B' \tag{13}$$

In the previous equation [74], it can be seen that R' and G' are important factors due to their high weight. In order to only compensate R and G to reduce computation. The human skin is characterized by a binary matrix based on Cr and experimental findings:

$$S_{ij} = \begin{cases} 0, 10 < C_r < 64 \\ 1, otherwise \end{cases} \tag{14}$$

where "0" represents the white and "1" represents the black point. In step two they used a 5×5 mask with a low pass filterto quickly remove high-frequency noise. Setting white points when the number of white points in a block of C_{ij} exceeds 50% of the total points; if the black points are greater thanhalf, change the blocks to black blocks. This filter success- fully filters out human skin. The third step identifying skin color blocks by storing four vertices of a rectangle around each region, identifying the

https://dx.doi.org/10.29322/IJSRP.14.01.2024.p14508

leftmost, rightmost, upmost, and lowermost points. This creates candidate blocks for detecting facial features.

In the final step detecting the height-to-width ratio is a quickand easy way to recognize human faces. For each candidate block, it finds characteristics like the height-to-breadth ratio, the mouth, and the eyes. This detection is given priority by low computation modules, and any candidate blocks with a height-to-width ratio between 1.5 and 0.8 are eliminated.

In Mouth Region Detection, the algorithm determines the height-to-width ratio of candidate blocks, morphs them based on image properties, and sorts of areas in descending order. It then finds the major difference in region and bounding box, which is crucial for mouth region detection. The algorithm calculates the centroid, and the meaning of these calculations, extracting the mouth region.

In Lip Region Detection, skin and lip pixels in RGB space have unique components, with red being more common and green being more frequent. Lips have a higher pseudo-hue than the skin, which is bijective. Choose the largest object and use the left and right corners of the mouth as the extremes to establish the lip form.

The result accuracy and realism are greatly enhanced because of its flexibility. This makes this approach ideal for tasks that need extreme accuracy, like lip reading. The model can replicate the uniqueness of the lips of a variety of speakers, as can be seen by the resulting lip shapes, which are extremely realistic and fit to the edges. Furthermore, the approach is reliable even in difficult situations like speakers with beards, inconsistent brightness, or when the teeth or tongue are visible. As long as the two comers are visible, segmentation can be done accurately even when the head is rotated. Such preliminary estimates are sometimes quite inaccurate. This project has several limitations. First, there is still a robustness concern with lip contour tracking. The majority of noisy motion vectors are reduced at the expense of ignoring some crucial motion data near the lip. Only the movement of the lip border pixels is taken into consideration while tracking the lip boundary over time. Also, since all of the photos were shot with seated participants and at the same distance, they only focused on the area around the lips [74].

Compared with the previous study for face recognition using lip features there are other papers that discuss other algorithms and results; some of them will be described. This study [75] investigates accurate lip-motion features for audio- visual speaker recognition, taking into consideration two basic representation options: 2D-DCT coefficients in the mouth region or lip boundary tracking over video frames. In the study, two alternative scenarios are taken into account: first, the rectangular mouth region is detected; second, mouth movement is represented by motion vectors, which may add irrelevant noisy motion vectors. The second scenario eliminates noisy motion vectors while still allowing for the inclusion of extracted lip shape information in the feature set. It does this by trackingthe lip boundary through time and only capturing the motionof lip border pixels. The audio-visual database MVGL-AVD is used for the biometric speaker identification tests; the database has 50 subjects. Every speaker in the database is represented with a different HMM (Hidden Markov Model) and a feature sequence that is retrieved from the lip stream as the secret phrase is spoken. Using the equal error rate (EER), the open- set speaker recognition system's performance is examined. The operational point where the false accept rate (FAR) and the false reject rate (FRR) are equal is known as the EER. The EER performance of the grid of size 64×40 is better thanthe results of the other grids.

$$FAR = 100 \times \frac{\text{number of false accepts}}{N_a + N_r} \qquad (15)$$

$$FRR = 100 \times \frac{\text{number of false rejects}}{N_a} \qquad (16)$$

Combining lip shape data with other available vectors im- proves EER performance. A different method for face detection and lip feature extraction is suggested in this research [76]. Also provided is a real-time field-programmable gate array (FPGA) implementation of the two suggested methods. A naive Bayes classifier that categorizes an image's edge- extracted representation provides the foundation for face detection. Lip feature extraction uses the contrast around the lip contour to extract the height and width of the mouth, metrics that are useful for speech filtering. By using edge representation, the model's size is significantly reduced, shrinking to only 5184 B, which is 2417 times smaller than a comparable statistical modeling technique. The findings showed that in various illumination situations, a proper detection rate of 86.6% could be achieved. Additionally, the same system that combined facial detection and lip feature extraction could tell if a speaker was speaking or not.

In contrast to current algorithms, the one presented in this paper [77] is more straightforward and does not require any prior training to detect human facial gestures. Based on lip structure research, the model recognizes three distinct facial gestures: happy, sad, and thinking. The algorithm's robustness was examined using several picture databases, including the authors' own database. Up to 95% of correct facial motions are produced by accurate lip detection. Although the model is not dependent on the orientation of the face, it occasionally produces incorrect results.

In other research [78], a new biometric identification systembased on lip shape measurements is proposed. This system consists of four steps: lips detection, envelope computation, envelope parameterization, and classifier recognition. A multi- labeled multiparameter hidden Markov model and a multilayer neural network are used to analyze the lip contour after it is extracted from a face color image, parameterized with polar and cartesian coordinates, and then rendered. The four primary phases in biometric recognition systems are database creation, feature extraction from samples, real-time sample comparison, and decision-making. The similarity of lips among individuals makes lip shape recognition difficult, but experimental results suggest that biometric identification by lip outline is feasible. To guarantee low intraclass variability, constraints must be applied. In the experiment, an average classification hit ratio of96.9% and an Equal Error Ratio (EER) of 0.015 were obtained using a database of 50 individuals.

H. A MATLAB-based face recognition system using image processing and neural networks

A novel method for recognizing human faces is presented in this research. By employing the two-dimensional discrete cosine transform (2D-DCT) to compress pictures and remove superfluous data from face photos, this method employs an image-based approach to artificial intelligence (2D-DCT) [79]. Based on skin color, the DCT derives characteristics from photos of faces. DCT coefficients are calculated to createfeature vectors. Moreover, to determine whether the object in the input picture is "present" or "not present" in the image dataset, DCT-based feature vectors are divided into groups using a self-organizing map (SOM), which uses an unsupervised learning method. By categorizing the intensity levels of grayscale images into several categories, SOM performs face recognition. Adding to these algorithms, MATLAB was used to evaluate the system [80].

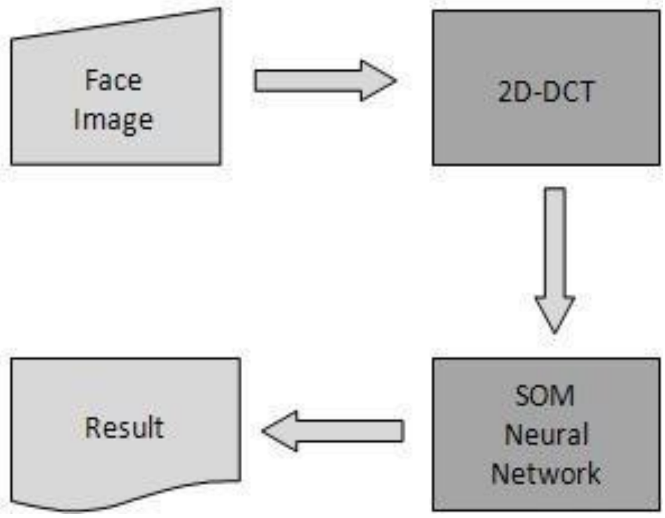


Fig. 12: Proposed technique for face recognition system

than chrominance. The 2D-DCT image compression method employs an intensity (grayscale) representation of the picture for further processing [81]. The second stage determines if the topic in the input image is "present" or "not present" in the image dataset by classifying vectors into groups usinga self-organizing map (SOM) and an unsupervised learning approach. If the subject is determined to be present, the training dataset's best-match image will be shown as the outcome.

The suggested design method uses "8" of the 64 DCT coefficients for masking to calculate the 2D-DCT of picture blocks with a size of 8 by 8 pixels. On the other side, one well-known artificial neural network is the self-organizing map, commonly referred to as a Kohonen Map. Without any class information, it is a technique of unsupervised learning that discovers the distribution of a set of patterns. The same method used by a competitive layer is used by a SOM network to choose a winning neuron. N nodes are arranged in a two- dimensional lattice form in the SOM network that is being employed here. The learning phase, the training phase, and the testing phase form the typical life cycle of a SOM.

Based on its recognition, the system's final output indicates whether the test picture is "present" or "not present" in the image dataset. Additionally, compared to conventional DCT feature extraction methods, the method's processing needs are drastically lowered by the use of a smaller feature space [80].

A different research study in [82] introduced a novel approach for identifying facial expressions. This method involvesutilizing a two-dimensional discrete cosine transform (DCT) across the facial image to identify distinctive features. They employed a constructive one-hidden-layer feedforward neural network to classify facial expressions. To enhance the learning process and decrease network size without compromising performance, they incorporated a technique called input-side pruning, which they had previously proposed.

They tested the technique using a dataset containing imagesof 60 men, each with five facial expression images (surprise, neutral, smile, sadness, anger). Out of them, 40 men's images were utilized for training the network, while the remaining 20 men's images were employed for testing and generalization. The performance of the trained network was assessed using confusion matrices for the four facial expressions (smile, anger, sadness, surprise) during both network training andgeneralization phases.

The findings revealed remarkable recognition rates: 100% for training images and 93.75% (excluding rejection). Additionally, the proposed technique reduced the input-side weights of the constructed network by approximately 30%. In comparison to existing fixed structure backpropagation- based recognition methods, this approach constructed a one- hidden-layer feedforward neural network with fewer hidden units and weights. This is achieved while simultaneously achieving enhanced capabilities in terms of generalization and recognition performance.

This research [83] proposes a human face detection algorithm for color photos based on the distribution and intensity of human skin colors. The intensity fluctuations and the skin color distribution that characterizes a human face are employed as features to divide the candidate regions into faces and non-faces. A non-linear luminance-based lighting compensation method is also used in the detecting process,and it is particularly effective at boosting and restoring the natural colors in pictures that were taken under a variety of different lighting conditions.

To quickly identify frontal human faces in challenging circumstances, technology has been created [84]. With a positive-negative attractor template and a valley detector for the eyes and mouths, it uses a quick picture segmentation technique. The neighbors that are on a pixel's up, down, left, and right sides were represented by four-neighbor pixels. With a database of 137 testers, the test results were 120 accurately detected, 17 missed, and 2 erroneously detected.

In other studies [85], liveness detection is essential for making sure a person is indeed present. Attackers must crop out the lips and eyes of the image to get around the liveness test, which leads to misclassification. The system makes use of the HSV value estimation, eye openness and proximity, and the HR classifier. Dimensionality reduction is accomplished using Principal Component Analysis (PCA). Using 40 test subjects and 20 photos, the system effectively recognizes all shown subjects and displays a high accuracy ratio.

In order to improve network performance, regularisation, and migration learning techniques are added, and GoogLeNet is enhanced to create the GoogLeNet-M network. This paper investigates the use of deep learning models in face recognition classification. According to experimental findings, the GoogLeNet-M network using regularisation and migration learning technologies performs best, with a recall rate and accuracy of 0.97 and 0.98, respectively [86].

I. *A Simple and Accurate Color Face Detection Algorithm in Complex Background.*

A quick face detection system with reliable results is described in this study. They take advantage of illumination adjustment to make color-based systems perform better and make feature-based systems less computationally difficult. Their approach works well on facial variations such as half- profile faces, phony faces, closed eyes, open mouths, and dark/bright eyesight. It is important to highlight that their algorithm can correctly identify between cartoon and human faces [87].

In paper [87], they suggest a face identification algorithm that can recognize a person's face in a variety of lighting situations. They distinguish between skin and non-skin regionsin the Y Cb Cr color space by using a color-based method. To address the issue of various lighting circumstances, they use nonlinear color transformation and lighting compensation techniques. They determine the facial features based on the height-to-breadth ratio of the face, human eyes, and lips. Where the white point is ”0” and the black point is ”1,” respectively, indicates the compensating effect on bright and dark images. They use a 5×5 mask to create a low pass filter in order to quickly filter out high-frequency noise. This 5×5 block is changed into a complete white block if there are more than half as many white points as black points, and vice versa for the black blocks. In order to mark skin color regions, they store vertices of rectangles for every region. For each candidate block, the features of height-to-width ratio, mouth, and eyes are consecutively recognized. After calculating the candidate blocks' height-to-width ratio. For each pixel in eachcandidate block, the value is determined. where ”0” designatesthe mouth pixel. To decide if there is a mouth in this block, they employ a vertical-based histogram. They used to measureit in a smaller area than the mouth for eye detection.

They used 300 static photos containing 700 real humanfaces and 20 fake faces to evaluate the suggested method. Twelve different camera brands, including Canon, Nikon, Sony, Fujifilm, and Olympus, were used to create these testimages. The image has a resolution of 320×240 pixels [87]. According to the experimental outcomes, their method detects frames in approximately 111 milliseconds in realtime. Additionally, out of 700 real faces applied, 646 weresuccessfully identified, and the suggested method's detectionrate is about 92.3%; their weakness was in unrecognizing 54 faces and 47 face detection errors [87].

J. *Design of an Efficient Face Recognition System Using Deep Learning Technique.*

In this research, face recognition has been implemented using Python and Arduino with real-time video in three phases, the initial phase includes identifying faces and gathering image IDs, the second of which involves training the recognizer and separating interesting elements, and the third of which includes grouping them and storing them in XML records[88]. The algorithm makes use of characteristics for edge or line detection. Phase 1 requires accessing the webcam, installing the computer vision modules using OpenCV, and then doing face detection. With the use of OpenCV's Cascade Classifier preparation method or pre-built models, it can be determined whether there is a face in the video transfer. There are two steps in the face detection process. The fundamental innovation is a data collection task that accepts a pair of contradictory images as data and returns a matched value of yes or no, indicating whether there are any faces present. The next advancement is the face limitation task, which aims to capture a photo for information. In order to store the data, they establish additional folders inside of the ”image data” folder [88]. Phase 2 involves getting the recognizer ready, taking care of the face information, and giving each face its own unique name so it can understand. Simultaneously, they stack the images to identify the faces in each image, a process called ”District of Interest”, and create a ”.yml” record with that information, and they accept all client data from the dataset and the OpenCV Recognizer at this stage. The creation of XML records to contain highlights extracted from datasets using the Face Recognizer class is made possible by Open CV. The face-recognizer class is used to create face-recognizer objects [88]. In Phase 3, face acknowledgment is included. Recognition involves taking care of unknown faces of people and determining whether the face recognizer is only equipped to recognize them. To identify faces that have been moved and managed to be seen in the image, a face identifier is used. Face Recognizer creates a figure for each face that is perceived. predict() provides the class ID and sureness as results. In addition to the software implementation, they related to hardware parts like Arduino interface with 16*2 LCD display [88]. The results are collected using software, and the status is presented on the LCD panel at the same time [88].

K. *Facial recognition using transfer learning in the deep CNN*

Transfer learning is one of the subcategories under machine learning that has drawn attention in the field of research over the past few years. Typical machine learning methods operate under the assumption that models train and test samples using a single-task distribution, while Transfer learning will address this by learning a new activity more effectively by transferring knowledge from a previously related task [89, 90].

The aim of this paper [91] is to study the utilization of transfer learning methods to enhance the effectiveness and precision of deep CNN-based facial recognition tasks [92]. Deep CNN, which is the Convolutional Neural Network isa widely used framework of deep learning. It is known that deep CNN has achieved exceptional outcomes in the field of face recognition. However, Deep CNN training can take a long time and requires a lot of labeled data, so this paper is investigating the efficiency of transfer learning deep CNNs for facial recognition tasks. Also, taking into consideration the effect of fine-tuning pre-trained models and using feature extraction in transferring knowledge from a source domain toa target domain.

The dataset in this study is done on two facial recognition datasets which are the subject of the experiment; Firstly, LFW (Labeled Faces in the Wild) contains 13,000 imagesof 5,749 people, and Secondly, CelebA, contains 202,599 images of 10,177 celebrities. For feature extraction and fine- tuning, two pre-trained CNN models are used; VGG-16 and ResNet-50. According to [93] VGG-16 is a convolution neural network (CNN) model supporting 16 layers. ResNET-50 is a convolutional neural network (CNN) supporting 50 layers, which is a vital network to be known [94]. Then, the results and performance of fine-tuning and feature extraction are compared with a linear classifier (they are often used as a baseline model to compare the performance of more complex models), and from 50 to 5,000 training samples are used at different times. The results showed that, Due to ResNet-50's more complex architecture, it did not perform as well after fine-tuning as VGG-16. It is crucial to select a trained model that is acceptable for the given facial recognition task.

Finally, the aim achieved as the study shows that transfer learning is an effective technique that may significantly improve the precision of facial recognition algorithms. Although feature extraction with a linear classifier outperformed fine-tuning previously trained CNN models in terms of accuracy, both methods succeeded in training from scratch [91].

L. Hardware implementation of Face Recognition systems

1) *Face Recognition System Based on Raspberry Pi Plat- form:* This paper [95] used Raspberry Pi to implement their work on face recognition. The Raspberry Pi is like a minicomputer, a smaller version of a modern computer that is able to execute a task efficiently. The module uses a varietyof processors, it can install open-source operating systems and applications. Also, Raspberry Pi supports various programming languages such as Python, C, and C++. Numerous applications in which the analysis of human facial expressions is crucial may utilize Raspberry Pi, which can detect real-time facial emotions [96, 97]. Also, using features provided by Python libraries such as OpenCV and NumPy. NumPy library stands for Numerical Python, used for scientific computingso it is crucial in terms of numerical calculations, consideredas the core of many other Python libraries that have derived from it, and it provides a high-performance multidimensional array object [98]. Raspberry Pi is the microprocessor used in this system. Besides, an integral part of its memory is SQLite. SQLite is the dataset management system used for storing the data of targeted people. The entire system process and the system architecture can be concluded in three stages starting with the dataset. From the Python code, the camera is ordered to take 20 sample pictures. Then, general data about the user's needs to be entered: name, age, gender, and department. All information is stored in the SQLite dataset in addition to the 20 sample pictures. Following this, pictures are updated if the data of the person is verified with an old record. The aim of the training section is to train the system to have the ability to recognize specific people. This is done by gray scaling. The Gray scaling format is used as it not only fastens and increases the efficiency of the algorithm of image processing but also lowers the data used. The detection process is the last section. The real-time video is transferred to the microprocessor by the camera. Then the video is imported as a two-dimensional matrix to the Python code, and the processing of the multidimensional matrix becomes easier with the use of the NumPy open-source library. The face is extracted from the full image and stored in a matrix. To determine any similarities, the matrix is compared with the sample pictures. For any similarities found, the name and further data of the recognized person that is stored in the SQLite database will be viewed on the real-time video plus a green square mark on the recognized face for tracking [95].

2) To enhance the system performance, the approach of face recognition works less appropriately when the recognition process is not running in the same condition as stored before. However, in this paper zero background effect is taken into consideration by extracting the entire face of the person from the stored images in the dataset; thus, it works appropriately even with any background or place changes. Another factor affecting the face recognition algorithm is the tiny changes like beards, make-up, and lighting conditions. The algorithm of this system examines the structure of the person and every differentedge part of the human face. Thus, the system is smartlydetecting the face throughout these tiny changes. The speed of processing of the face recognition system is a crucial part. In this paper, images are saved in the dataset in XML format. XML (Extensible Markup Language) file is used rather than a PNG (Portable Network Graphic) file, as XML is easier and does not consume time during the process. Besides, the image processing takes a shorter time as, throughout the process, onlythe face is extracted from the image so other information in theimage is not processed. Another integral factor for decreasing the time is the gray scaling. As a result, four frames per secondare processed by the system. On the other hand, identifying a moving person through a slow image recognition algorithm is a challenge faced in this paper [95].

II. CONCLUSION

Face recognition is a vital research topic in the image processing and computer vision fields because of both its theoretical and practical impact. Access control, imae search, human-machine interfaces, security, and entertainment are just a few of the real-world applications for face recognition. However, these applications confront several difficulties, including lighting conditions. Numerous research articles on both software and hardware are highlighted in this document as well as a deep dig into the algorithm's comprehension, concentrating mostly on methods based on local, holistic, and hybrid features.

REFERENCES

[1] ChillWall, "Future of facial recognition technology," December 2020. [Online]. Available: https://blog.chillwall.com/2020/12/04/facial- recognition-technology/

[2] V. Sati, S. Márquez-Sánchez, N. Shoeibi, A. Arora, and J. Cor- chado Rodr´ıguez, "Face detection and recognition, face emotion recog- nition through nvidia jetson nano," pp. 177–185, 09 2020.

[3] S. Liao, A. K. Jain, and S. Z. Li, "Partial face recognition: Alignment- free approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1193–1205, 2013.

[4] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220316945

[5] "Usd 12.92 billion by 2027; increasing demand for advanced video surveillance systems to augur growth: Fortune business insightstm," July 2020. [Online]. Available: https://www.globenewswire.com/en/news-release/2020/07/09/2059692/0/en/Facial-Recognition-Market-to-Reach- USD-12-92-Billion-by-2027-Increasing-Demand-for-Advanced-Video-Surveillance-Systems-to-Augur-Growth-Fortune-Business-InsightsTM.html

[6] W. Crumpler and J. A. Lewis, "How does facial recognition work?" The Center for Strategic and International Studies, Tech. Rep., 2021.

[7] Z. Akhtar and A. Rattani, "A face in any form: New challenges and opportunities for face recognition technology," *Computer*, vol. 50, no. 4, p. 80–90, 2017.

[8] D. N. Parmar and B. B. Mehta, "Face recognition methods applications," *Int. J. Computer Technology Applications*, pp. 84–86, 2013.

[9] R. Jafri and H. R. Arabnia, "A survey of face recognition techniques," *Journal of Information Processing Systems*, vol. 5, no. 2, pp. 41–68, 2009.

[10] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recogni- tion: A literature survey," *ACM Comput. Surv.*, vol. 35, no. 4, p. 399–458, 2003.

[11] "Innovation that matters," August 2019. [Online]. Available: https://www.springwise.com/tech-explained/facial-recognition

[12] S. Kummathi, S. Reddy, and K. Nagabhushan Raju, "Design and implementation of an algorithm for face recognition by using principal component analysis (pca) in matlab," pp. 115–119, 10 2016.

[13] C. A. Hansen, "Face recognition," *Institute for Computer Science University of Tromso*.

[14] J. S. Rambey, N. Sinaga, and B. D. Waluyo, "Automatic door access system using face recognition," *International Journal of Computer Applications Technology and Research*, p. 371–376, 2022.

[15] M. Lal, K. Kumar, R. Hussain, A. Maitlo, S. Ali, and H. Shaikh, "Study of face recognition techniques: A survey," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 6, 2018.

[16] S. Ohlyan, S. Sangwan, and T. Ahuja, "A survey on various problems challenges in face recognition," *International journal of engineering research and technology*, vol. 2, 2013.

[17] S. M. K. Hasan and T. Chowdhury, "Face recognition using artificial neural networks," 03 2004.

[18] M. Aneesa, N. Sabina, and K. Meera, "Face recognition using cnn: A systematic review," *International Journal of Engineering Research Technology*, vol. 11, no. 6, pp. 182–185, 2022.

[19] M. M. Kasar, D. Bhattacharyya, and T.-H. Kim, "Face recognition using neural network: A review," *Int. Journal of Security and Its Applications*, vol. 10, no. 3, pp. 81–100, 2016.

[20] A. A. Al-blushi, "Face recognition based on artificial neural network: A review," *Artificial Intelligence Robotics Development Journal*, vol. 1, no. 2, pp. 116–131, 2021.

[21] J. Wang and Z. Li, "Research on face recognition based on cnn," ser. 170. IOP, 2018, p. 5.

[22] R. E. Saragih and Q. H. To, "A survey of face recognition based on convolutional neural network," *Indonesian Journal of Information Systems*, vol. 4, no. 2, pp. 122–139, 2022.

[23] P. Purwono, A. Ma'arif, W. Rahmaniar, H. Imam, H. I. K. Fathurrahman, Z. Frisky, and Q. M. U. Haq, "Understanding of convolutional neural network (cnn): A review," *International Journal of Robotics and Control Systems*, vol. 2, pp. 739–748, 2023.

[24] Mathworks. (2018) Introducing deep learning with matlab. [Online]. Available: https://www.mathworks.com/campaigns/offers/next/deep-learning-ebook.html3

[25] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into Imaging*, vol. 9, no. 4, pp. 611–629, 2018.

[26] M. M. Hussein, A. H. Mutlag1, and H. Shareef, "An improved artificial neural network design for face recognition utilizing harmony search algorithm," vol. 745. IOP Conf. Series: Materials Science and En-gineering, 2020.

[27] L. Alzubaidi, J. Zhang, A. Humaidi, Amjad J.and Al-Dujaili, Y. Duan, O. Al-Shamma, M. A. A.-A. M. Santamar´ıa, J.and Fadhel, and L. Farhan, "Review of deep learning: concepts, cnn architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, pp. 53–127, 2021.

[28] P. Kamencay, M. Benco, T. Mizdos, and R. Radil, "A new method for face recognition usingconvolutional neural network," *DIGITAL IMAGE PROCESSING AND COMPUTER GRAPHICS*, vol. 15, no. 4, pp. 663– 672, 2017.

[29] Y. Said, M. Barr, and H. E. Ahmed, "Design of a face recognition system based on convolutional neural network (cnn)," *Engineering, Technology Applied Science Research*, vol. 10, no. 3, pp. 5608–5612, 2020.

[30] "Att database of faces: Orl face database." [Online]. Available: http://cam-orl.co.uk/facedatabase.html

[31] A. B. S. Salamh and H. I. Akyuz, “A new deep learning model for face recognition and registration in distance learning,” *Int. Journal of Emerging Technologies in Learning*, vol. 17, no. 12, pp. 29–41, 2022.

[32] S. Karamizadeh, S. M. Abdullah, A. A. Manaf, M. Zamani, and A. Hooman, “An overview of principal component analysis,” *Journal of Signal and Information Processing*, vol. 4, pp. 173–175, 2013.

[33] C. Li, Y. Diao, H. Ma, and Y. Li, “A statistical pca method for face recognition,” vol. 3, pp. 376–380, 2008.

[34] L. Sirovich and M. Kirby, “Low-dimensional procedure for the charac- terization of human faces,” *Journal of the Optical Society of AmericaA*, vol. 4, no. 3, pp. 519–524.

[35] L. Paul and A. Suman, “Face recognition using principal component analysis method,” *International Journal of Advanced Research in Computer Engineering Technology (IJARCET)*, vol. 1, pp. 135–139, 11 2012.

[36] H. Duan, R. Yan, and K. Lin, “Research on face recognition based on PCA,” pp. 29–32, 2008.

[37] Z. Wang and X. Li, “Face recognition based on improved PCA reconstruction,” in *2010 8th World Congress on Intelligent Control and Automation*, 2010, pp. 6272–6276.

[38] D. S. Pankaj and M. Wilscy, “Comparison of PCA, LDA and gabor features for face recognition using fuzzy neural network,” 2012.

[39] F. R. U. P. C. A. Method, “A face recognition system based on eigenfaces method,” *Elsevier*, vol. 1, no. 9, pp. 135–139, 2012. [Online]. Available: 544-1569-1-PB-libre.pdf (d1wqtxts1xzle7.cloudfront.net)

[40] M. Ç arıkçı and F. Ö zen, “A face recognition system based on eigenfaces method,” *Elsevier*, vol. 1, pp. 118–123, 2012.

[41] T. F. Karim, M. S. H. Lipu, M. L. Rahman, and F. Sultana, “Face recognition using PCA-based method,” in *2010 IEEE International Conference on Advanced Management Science (ICAMS 2010)*, vol. 3, 2010, pp. 158–162.

[42] K. Teoh, R. Ismail, S. Naziri, R. Hussin, M. Isa, and M. Basir, “Face recognition and identification using deep learning approach,” *Journal of Physics: Conference Series*, vol. 1755, 2021.

[43] B. Benuwa, Y. Zhan, B. Ghansah, D. Wornyo, and F. Banaseka, “A review of deep machine learning,” *International Journal of Engineering Research in Africa*, vol. 24, pp. 124–136, 06 2016.

[44] I. Gupta, V. Patil, C. Kadam, and S. Dumbre, “Face detection andrecognition using raspberry pi,” 12 2016.

[45] P. S. Prasad, R. Pathak, V. K. Gunjan, and H. V. Ramana Rao, “Deep learning-based representation for face recognition,” *Lecture Notes in Electrical Engineering*, p. 419–424, 2019.

[46] J. Mahreen, “Working with images and videos using opencv,” *Analytics Vidhya*, Aug 2022.

[47] I. uljak, D. Abram, D. Pribanić, H. Dapo, and H. Cifrek, “A brief introduction to OPENCV,” *2012 Proceedings of the 35th International Convention MIPRO*, pp. 1725–1730, 2012. [Online]. Available: https://api.semanticscholar.org/CorpusID:7678499

[48] J. Vadlapati, S. Senthil Velan, and E. Varghese, “Facial recognition using the OPENCVs libraries of python for the pictures of human faces wearing face masks during the covid-19 pandemic,” *2021 12th International Conference on Computing Communication and Networking Technologies(ICCCNT)*, 2021.

[49] H. Adusumalli, D. Kalyani, R. Sri, M. Pratapteja, and P. Rao, “Face mask detection using opencv,” pp. 1304–1309, 02 2021.

[50] S. Ahlawat, *Introduction to TensorFlow*. Berkeley, CA: Apress, 2023, pp. 5–137.

[51] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” *Proceedings of the 22nd ACM international conference on Multimedia*, 2014.

[52] J. Joseph, F. Joe, S. Nonsiri, and A. Monsakul, *Keras and TensorFlow: A Hands-On Experience*, 07 2021, pp. 85–111.

[53] L. Quaranta, F. Calefato, and F. Lanubile, “Kgtorrent: A dataset of python jupyter notebooks from kaggle,” *2021 IEEE/ACM 18th Inter- national Conference on Mining Software Repositories (MSR)*, 2021.

[54] Uslu, “What is kaggle?” *Data Camp*, 2022.

[55] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution grayscale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, p. 971–987, 2002.

[56] M. PietikÃ ¤inen, “Local binary patterns,” *Scholarpedia*, vol. 5, no. 3, p. 9775, 2010.

[57] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, “Local binary patterns and its application to facial image analysis: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, p. 765–781, 2011.

[58] T. Ahonen, A. Hadid, and M. Pietikäinen, “Face recognition with local binary patterns,” vol. 3021, 05 2004, pp. 469–481.

[59] A. Salihbasic and T. Orehovacki, “Development of android application for gender, age and face recognition using opencv,” *2019 42nd Inter- national Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2019.

[60] F. Deeba, H. Memon, F. Ali, A. Ahmed, and A. Ghaffar, “Lbph-based enhanced real-time face recognition,” *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 5, 2019.

[61] M. Karaaba, L. Surinta, Olarik Schomaker, and M. Wiering, “Robust face recognition by computing distances from multiple histograms of oriented gradients,” ser. 1. IEEE Symposium Series on Computational Intelligence, 2015, p. 7.

[62] J. Thilaga and P. Ayyadurai, “Modern face recognition with deep learning,” *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, 2018.

[63] S. Qutbuddin and S. Larik, “Face recognition for automated attendance using hog machine learning,” ser. 1. ResearchGate, 2021, p. 4.

[64] A. B. M Maria Dominic Savio, T Deepa and T. S. Anurag, “Image processing for face recognition using haar, hog, and svm algorithms,” ser. 1964. IOP, 2021, p. 9.

[65] C. Xuefeng, F. Long, and C. Huang, “Face recognition by zero-ratio based lgbp features,” ser. 1. 11th World Congress on Intelligent Control and Automation (WCICA), 2014, pp. 5605–5608.

[66] P. Viola and M. Jones, “Rapid object detection using a

boosted cascade of simple features," ser. 1. ResearchGate, 2004, p. 12.

[67] ——, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 1, no. 2, pp. 120–125, 2004.

[68] C. Huang and J. Huang, "A fast hog descriptor using lookup table and integral image," 03 2017.

[69] A. M. J. S. Alberto Albiol, David Monzo and A. Albiol, "Face recognition using hog-ebgm," *Elsevier*, vol. 29, no. 1, pp. 1537–1543, 2008.

[70] L. A. G. E. T. M. Bicego, M., "On the use of sift features for face authentication," in *on Computer Vision and Pattern Recognition Workshop, New York.*, vol. 3, 2006, pp. 158–162.

[71] D. Lowe, "Distinctive image features from scale-invariant keypoints." *International Journal of Computer*, vol. 60, no. 1, pp. 91–110, 2004.

[72] J. M. Viola, P., "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 2, 2001.

[73] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European Conference on Computer Vision*, 2006.

[74] S. Gurumurthy and B. K. Tripathy, "Design and implementation of face recognition system in matlab using the features of lips." *International Journal of Intelligent Systems and Applications (IJISA)*, vol. 4, no. 8, pp. 30–36, July 2012.

[75] H. Cetingul, Y. Yemez, E. Erzin, and A. Tekalp, "Robust lip-motion features for speaker identification," *Acoustics, Speech, and Signal Pro- cessing, 1988. ICASSP-88., 1988 International Conference on*, vol. 1, pp. 509–512, 02 2005.

[76] D. Nguyen, D. Halupka, P. Aarabi, and A. Sheikholeslami, "Real-time face detection and lip feature extraction using field-programmable gate arrays," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 36, pp. 902 – 912, 09 2006.

[77] J. Raheja, R. Shyam, J. Gupta, U. Kumar, and P. Prasad, "Facial gesture identification using lip contours," *ICMLC 2010 - The 2nd International Conference on Machine Learning and Computing*, pp. 3 – 7, 03 2010.

[78] E. Gomez, C. Travieso, J. Briceno, and M. Ferrer, "Biometric identification system by lip shape," *IEEE Annual International Carnahan Conference on Security Technology, Proceedings*, pp. 39 – 42, 02 2002.

[79] L. Ma and K. Khorasani, "Facial expression recognition using constructive neural networks," 2001.

[80] J. Nagi, S. Ahmed, and F. Nagi, "A MATLAB-based face recognition system using image processing and neural networks," January 2008.

[81] A. R. Chadha, P. P. Vaidya, and M. M. Roja, "Face recognition using discrete cosine transform for global and local features," pp. 502–505, 2011.

[82] L. Ma and K. Khorasani, "Facial expression recognition using constructive feedforward neural networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 3, pp. 1588–1595, 2004.

[83] S. Gundimada, l. Tao, and V. Asari, "Face detection technique based on intensity and skin color distribution," vol. 2, pp. 1413 – 1416 Vol.2, 11 2004.

[84] Q. Yuan, W. Gao, and H. Yao, "Robust frontal face detection in complexenvironment," vol. 1, pp. 25 – 28 vol.1, 02 2002.

[85] A. Singh, P. Joshi, and G. Nandi, "Face recognition with liveness detection using eye and mouth movement," *2014 International Conference on Signal Propagation and Computer Technology, ICSPCT 2014*, 07 2014.

[86] Z. Yu, Y. Dong, J. Cheng, M. Sun, and F. Su, "Research on face recognition classification based on improved GoogleNet," *Security and Communication Networks*, vol. 2022, p. 7192306, Jan 2022. [Online]. Available: https://doi.org/10.1155/2022/7192306

[87] Y.-T. Pai, S.-J. Ruan, M.-C. Shie, and M.-C. Liu, "A simple and accurate color face detection algorithm in complex background." *Low Power Systems Lab, Department of Electronic Engineering National Taiwan University of Science and Technology.*, no. 43, 2006. [Online]. Available: https://www.cecs.uci.edu

[88] P. Rathnala, T. M. P. K. Patnaik, S. R. Sura, B. Prasad, N. S. M. Rao, and D. N. Rayan, "Design of an efficient face recognition system using deep learning technique." *International Journal of Electrical and Electronics Research (IJEER)*, vol. 10, pp. 689–693, 2022. [Online]. Available: https://ijeer.forexjournal.co.in/archive/volume-10/ijeer-100345.html

[89] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: A friendly introduction," *Journal of Big Data*, vol. 9,no. 1, 2022.

[90] L. Torrey and J. Shavlik, *Transfer learning - university of wis- consin–madison*. [Online]. Available: https://ftp.cs.wisc.edu/machine- learning/shavlik group/torrey.handbook09.pdf

[91] H. O. Ikromovich and B. B. Mamatkulovich, "Facial recognition using transfer learning in the deep cnn," *Open Access Repository*. [Online]. Available: https://oarepo.org/index.php/oa/article/view/2264

[92] N. Aloysius and G. Madathilkulangara, "A review on deep convolutionalneural networks," pp. 0588–0592, 04 2017.

[93] S. Tammina, "Transfer learning using vgg-16 with deep convolutional neural network for classifying images," *International Journal of Scientific and Research Publications (IJSRP)*, vol. 9, p. p9420, 10 2019.

[94] B. Koonce, *ResNet 50*, 01 2021, pp. 63–72.

[95] N. Mustakim, N. Hossain, M. M. Rahman, N. Islam, Z. H. Sayem,and M. A. Mamun, "Face recognition system based on raspberry pi platform," *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, 2019.

[96] H. Ghael, "A review paper on raspberry pi and its applications," 01 2020.

[97] N. Rathour, Z. Khanam, A. Gehlot, R. Singh, M. Rashid, A. S. Alghamdi, and S. S. Alshamrani, "Real-time facial emotion recognition framework for employees of organizations using raspberry-pi," 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/22/10540

[98] F. Nelli, *The NumPy Library*, 01 2015, pp. 35-61.