Face Recognition Using Principal Component Analysis, K-Means Clustering, and Convolutional Neural Network

Zukisa Nante, Wang Zenghui

Abstract**—**Face recognition is the problem of identifying or recognizing individuals in an image. This paper investigates a possible method to bring a solution to this problem. The method proposes an amalgamation of Principal Component Analysis (PCA), K-Means clustering, and Convolutional Neural Network (CNN) for a face recognition system. It is trained and evaluated using the ORL dataset. This dataset consists of 400 different faces with 40 classes of 10 face images per class. Firstly, PCA enabled the usage of a smaller network. This reduces the training time of the CNN. Thus, we get rid of the redundancy and preserve the variance with a smaller number of coefficients. Secondly, the K-Means clustering model is trained using the compressed PCA obtained data which select the K-Means clustering centers with better characteristics. Lastly, the K-Means characteristics or features are an initial value of the CNN and act as input data. The accuracy and the performance of the proposed method were tested in comparison to other Face Recognition (FR) techniques namely PCA, Support Vector Machine (SVM), as well as K-Nearest Neighbour (kNN). During experimentation, the accuracy and the performance of our suggested method after 90 epochs achieved the highest performance: 99% accuracy F1-Score, 99% precision, and 99% recall in 463.934 seconds. It outperformed the PCA that obtained 97% and KNN with 84% during the conducted experiments. Therefore, this method proved to be efficient in identifying faces in the images.

Keywords**—**Face recognition, Principal Component Analysis, PCA, Convolutional Neural Network, CNN, Rectified Linear Unit, ReLU, feature extraction.

I. Introduction

THE idea behind the FR system is to give computers the ability to find, detect, and recognize human faces accurately. This can be performed in images and videos. In recent years a lot of researchers, including Bai et al. [17], extensively researched the areas of pattern recognition, signal processing, and computer vision. Deep learning computer vision algorithms, methods, and techniques have shown improvement in FR systems. Humans use faces to identify known and unknown faces since the beginning of civilization. This is done effortlessly as the human brain can instantly and automatically recognize familiar and non-familiar faces. FR systems can be found in security systems at the airport, by the police, for criminal identification and verification systems, etc. FR is a very interesting biometric modality as it is the natural mode of identification amongst humans and is very unobtrusive. As the name suggests the process of FR happens when a face of a person is recognized. A FR system comprises four stages:

- Face Detection - detects the localization of the image, verifies if a face or faces exist(s) in an image and if it does it draws a bounding box on the face (see Fig. 1).
- Face Alignment – is normalization of a face to be exact and comparable with the database format such as photometric and geometry.
- Feature Extraction – carefully extracts the usable face features to help during the recognition assignment.
- FR - compares these features from the faces in the database, verifies if a match exists and if it does it recognizes that person by assigning a label trained on it.

Face detection and recognition differ in the sense that in detection the interest is only to know if a face exists from an image or static picture or video, but recognition task is a procedure of recognizing an already detected face or identifying who the person is [1]. Nair and Hinton [5] define object recognition as the way to keep or maintain the same input properties in the output invariance. FR systems work by comparing selected facial features from a given image with faces within a database. However, these systems still face problems of illumination, pose variation, expression changes, and facial disguises. Different illumination conditions affect the detection and recognition accuracy. It is observed that detecting faces using the Haar cascade classifier restricts poses that may hinder the accuracy when feature extraction is performed. This classifier detects a face when certain illumination conditions are met. It draws a bounding box to confirm the localization of a face.

Image quality, head orientation, lighting conditions, partial occlusion, and facial expressions play an important role during feature extraction [1]. The extraction of meaningful features is a very important task, especially in FR, thus, a feature-based system speeds up the process more than a pixel-based system [8]. FR techniques use features like mouth, eyes, chin, nose, and geometrically assess relationships among them. Zhao et al. [15] suggested a FR system utilizing a deep neural network with PCA, jointly with Bayesian framework, and achieved 98.52% performance from their dataset, which is the CAS-PEAL dataset. Bhaskar et al. [2] proposed a FR system based on Hybrid Gaborlet and Fisher Analysis to overcome the problem of the pose, illumination, age, occlusion, and expressions where the

Zukisa Nante and Wang Zenghui are with the Department of Electrical Engineering, University of South Africa, Florida 1709, South Africa (e-mail: 37229958@mylife.unisa.ac.za, wangz@unisa.ac.za).

This research is supported partially by South African National Research Foundation Grants (No. 112108, 132797, and 137951), and Tertiary Education Support Program (TESP) of South African ESKOM.

results were exceptional using Dr. Libor Spacek (800 face images of 40 classes with 20 views/class) and Caltech (360 face images with 18 classes with 20 views/class) segmented databases. Ren et al. [9] highlighted that the Region Proposal Network (RPN) reveals the whole convolution features of the image to a detection network.

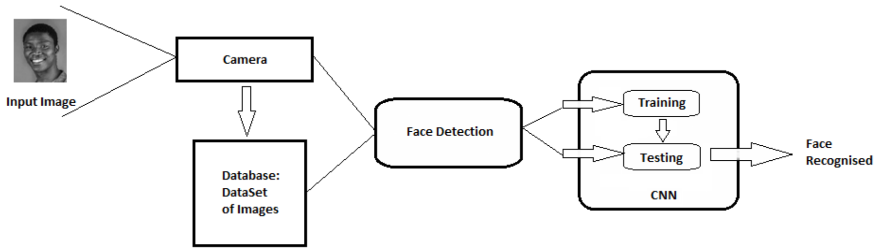

Fig. 1 Face Recognition system

In this research, we attempt to address the face feature extraction problem, so that we can preserve good features to obtain better results by combining PCA, K-Means clustering, and CNN architecture. This is attempted by analyzing the existing methods, extracting features, observing the accuracy of each, and combining the methods that promised to improve the performance and accuracy. To deal with this problem, in an attempt to reach a solution, we discuss the following FR methods:

- Neural Network (NN): CNNs easily recognize normalized and aligned faces and increase the efficiency of the model.
- Kernel PCA: This methodology applies nonlinear mapping to the input image and then solves linear PCA in the resulting feature subspace. This technique can help in dimensionality reduction; however, it agonizes from high computational overload.
- Geometric Based: This method analyses local facial features and their relationship. It can be constructed with tools like PCA and SVM.
- K-Means Clustering: This one groups samples based on their feature similarities. While this algorithm is exceptional at recognizing clusters with a spherical space, it has a drawback, and that is we must give the number of clusters, k, a priori.

The paper outline is arranged in the following manner: Section II contends with the problem definition, fundamental concepts, proposed contributions, and methodology applied in this paper. Section III discusses training and simulations. Section IV lists the obtained results. Section V is conclusion and future direction.

II. Proposed Methodology

The proposed technique namely Improved Deep Neural Network (IDNN) comprises Deep Convolutional Neural Networks (DCNNs) models which are tested for FR as they allow extraction of a wide range of features from images. It utilizes PCA first for dimensionality reduction and pre-processing and K-Means clustering before it enters the DCNN. This is discussed in detail in the following subsections, but we first discuss the problem and fundamental concepts behind it.

The primary problem definition of a FR system is the ability to recognize a person's face from an image in a set of testing images from a dataset in the database. Illumination, pose, and facial expression affect the process of feature extraction and selection. Hence, de-noising the image with a filter helps preserve appropriate features such as edges to enhance accuracy during the training process. It is crucial to fully understand the essential concepts behind FR to effectively comprehend the proposed technique. Pre-processing is discussed, Feature Extraction (FE) using PCA is evaluated. SVM classification is considered for its effectiveness in high dimensional spaces.

A. Image Pre-processing

Image pre-processing is a vital step to make sure that raw data is of the required form and shape. The aim is to improve image data by suppressing unwanted or undesired distortions and enhancing relevance in image features for further processing. In our case, the sequence of face images that we seek to take out significant attributes namely, the outline from the eye sockets, nose, mouth, as well as chin are raw data. Therefore, image pre-processing helps in feature transformation to a gamut of [zero, one] or a traditional regular allocation of a 0 mean with unit variation [11].

B. FE Using PCA

FE is used to convert or launch information onto a new characterized collection of features. However, when used to reduce the dimensions the purpose is to compress data and highly maintain the relevant features. Therefore, the predictive performance is enhanced by reducing curse dimensionality. Richard Bellman, 1957, defines curse dimensionality as an exponential in volume associated with adding extra dimensions to Euclidean space [12]. This means an error increases with the increase in the number of features. Here, FE and dimensionality reduction are performed by utilizing the PCA as it helps in distinguishing models within our data correlating between facial attributes. It seeks to allocate a path of the largest variant in high-level dimensionality data, then projects it directly to the newest subspace of equivalent or lesser dimensions compared to the first images. Hence, PCA dimension reduction constructs the d x k-dimensional conversion matrix W which permits mapping of the test value x against a recent k-dimensional attribute vector space with fewer dimensions than the initial d-dimensional attribute space such that:

$$x = [\mathrm{x}_1, \mathrm{x}_2, \dots, \mathrm{x}_\mathrm{d}], x \in \mathbb{R}^d \quad (1)$$

$$\downarrow \quad xW, W\, x \in \mathbb{R}^{d\times k}$$

$$z = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_\mathrm{k}], x \in \mathbb{R}^{k} \tag{2}$$

C. Using SVM

SVM perceptron algorithm is used to minimize misclassification errors. However, we use it to maximize the image margins. Here, the margin is the space amongst the splitting hyperplane and training small parts close to the hyperplane. Hyperplanes or decision boundaries with large margins tend to have a lower generalization error while models with smaller margins are more susceptible to overfitting. The basic equation behind the margin maximization is given by (3):

$$\frac{w^T(x_{pos}-x_{neg})}{||w||} = \frac{2}{||w||} \tag{3}$$

where w is the vector length, x_{pos} denotes a positive hyperplane, and x_{neg} a negative hyperplane. The left side of (3) signifies the margin to maximize which is the distance between the positive and negative hyperplanes. The SVM function becomes the maximization of this margin by maximizing the righten side of the equation on constraints that the classification of samples is correct in the dataset.

The following subsections as mentioned at the beginning of this section are to elaborate and illustrate the proposed contributions of our proposed methodology.

D. Convolutional Neural Networks

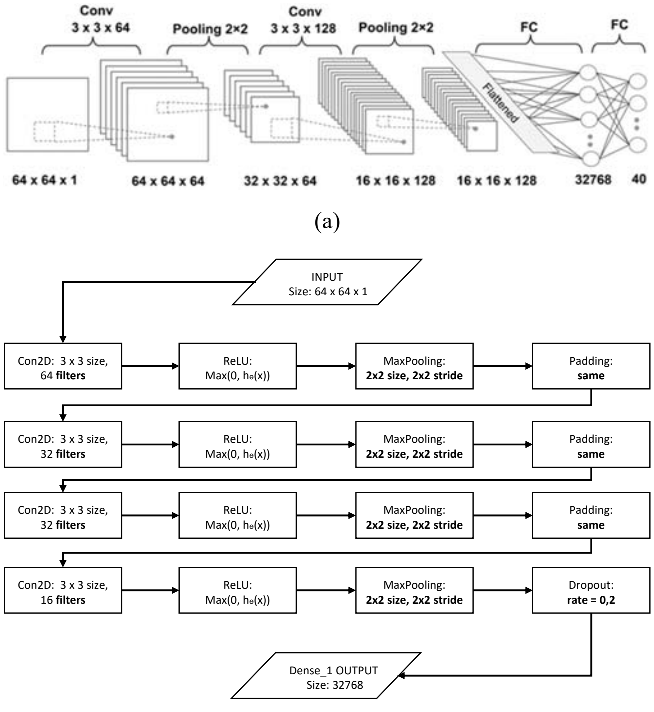

Fig. 2 The CNN architecture used in this paper: (a) the complete architecture and (b) the layers in detail

CNN models were motivated by the fundamental working cortex of the human brain when recognizing objects. Their evolution originated in the 1990s after Yann LeCun and co-workers suggested a new different NN structure intended for handwritten numbers classification from images [17]. They are NNs that are composed of a mathematical operation called convolution, hence the name CNN. Their ability to spontaneously learn features from unprocessed data makes pre-processing easy in comparison to other image classification algorithms mentioned in Section I. They construct a collection of features by merging minor details of the image like edges in the form of layers to form recognizable features. They compute feature maps from the input data and in our case from the face image, in which, every single essential part originates from a neighboring patch of pixels. This local patch is called the local receptive field [3]. CNNs result from neurons containing weights and biases. Their goal is to go from the unprocessed input data in the first layer to the exact class in the last layer. They differ from normal NNs due to the type of layers they use and the way they treat the input data. They assume input data like images and that allows them to extract features specific to the images [10]. Fig. 2 illustrates the proposed multilayer CNN architecture for this paper.

Fig. 3 depicts the suggested method flowchart.

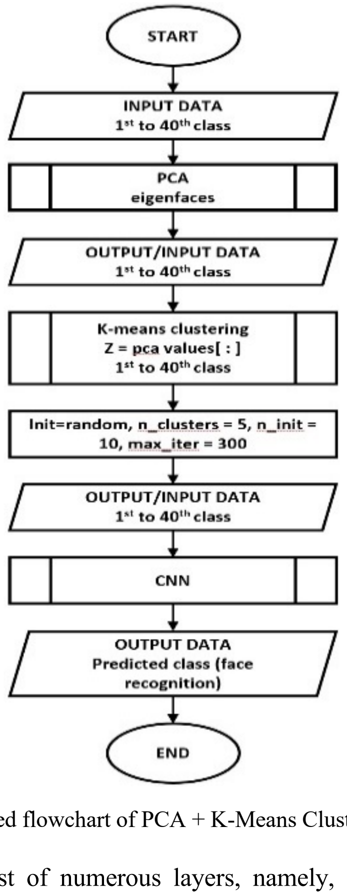

Fig. 3 Suggested flowchart of PCA + K-Means Clustering + CNN

CNNs consist of numerous layers, namely, convolutional, Pooling (P), and Fully Connected (FC) layers. Each layer transforms its volume of Rectified Linear Unit (ReLU) activations to another layer through different functions. They are also comprised of a dropout layer to avoid overfitting, [6]. Here, we have incorporated three main types of layers which are convolutional layers, P layers, and FC layers. The layers of this architecture are illustrated as:

- Input layer consists of raw image data with a dimension vector of batch size 64 x 64 x 1. This means: batch size x height x width x grayscale images (1 channel images).
- Convolutional layer (Conv_1) entails a batch size x 64 x 64 x 64 with a kernel of 3 x 3 and 64 output feature maps. This layer computes the convolutions between the neurons and the various features in the input.
- Pooling layer (Pooling_1) consists of a batch size x 32 x 32 x 64 where (32 x 32) represents the height and width and 64 representing output feature maps. This layer samples the previous layer and results in reduced dimensions. This layer focuses on keeping important features and we use Max pooling to keep maximum value in each *K x K* window. Max pooling presents some sort of local invariance, consequently, it helps to produce vigorous features against noise in the input data. Here, the same padding is utilized to get the same input and output size of *x*. By doing so we compute the padding parameter, p, corresponding to the filter size to make sure that the input size is the same as the output size as required [3].
- ReLU is an activation function that is applied to the matrix to make it linear [3], [4]. It is linear in the positive dimension but not in the negative dimension. Its linearity in the positive dimension has an attractive feature that inhibits non-saturation of gradients, though one-half of the actual line its gradient is zero. It takes the summed weighted input from the node and transforms it into the activation of the node or output for that input.
- Convolutional layer (Conv_2) entails a batchsize x 16 x 16 x 128 with a kernel of 3 x 3 and 128 output feature maps.
- Pooling layer (Pooling_2) consists of a batch size x 16 x 16 x 128 with a kernel of 16 x 16 and 128 representing output feature maps.
- Fully Connected (FC_2) and softmax layers calculate the output score emerging in size up to 1 x 1 x 40 where 40 defines the number of classes in the training dataset. According to [7], this layer can be trained end-to-end, pixels-to-pixels on semantic image segmentation (classification and localization) rather than only to predict the dense output.

E. Image Classification Using CNN

Classifying images with deep CNNs requires the use of TensorFlow. The CNN FC layers are fundamentally relating to several layers perceptron whereby each input unit *i* is connected to each output unit *j* together with weight *wij*. Multilayer NNs in CNNs construct a system of features by merging the features of the blobs and edges in layers formulating events and object features. In two dimensions (2D) of X_{n1xn2} input together with a square matrix W_{m1xm2} whereby m1 <= n1 and m2 <= n2, so the matrix Y = X * W outcome is the 2D convolution of X as well as W. It can be calculated as:

$$y = X * W \rightarrow Y[i, j] = \sum_{k1=-\infty\cdot}^{+\infty} \sum_{k_2=-\infty}^{+\infty} X[i - k_1, j - k_2] W[k_1, k_2] \quad (4)$$

F. PCA for Data Compression and Classification

Using PCA as an unsupervised dimensionality reduction we utilize a FE algorithm to convert or project data onto a new feature space to maintain most of the relevant information and this is accomplished by using eigenvalues and eigenvectors. Thus, this improves the computing time that the algorithm learns enormously by decreasing the dimensionality curse. To understand the training algorithm of eigenfaces, we consider an individual's facial representation of I(x, y) as a 2D N x N ordered arrangement. Fig. 4 displays the original face images to demonstrate the calculation of Eigenfaces.

Fig. 4 ORL faces Dataset [16]

These images are converted into vectors of size N^2 so that an image of 64 x 64 turns into a vector size of 4096 dimensions or equivalent to a point in a 4096-space dimension. The main purpose is to get vectors that at a higher leve, consider the dissemination of the faces in the whole space of an image. These faces have identical similarities and the variance equivalent to initial images are called eigenfaces [13]. Assuming a collection of images to be trained is $x_1, x_2, x_3 \ldots x_m$ we can compute the median of all these face vectors by

$$\Psi = \frac{1}{M} \sum_{n=1}^{M} x_1 \quad (5)$$

and because each face varies from the average face Ψ we can subtract it by

$$\Psi = c - \Psi \quad (6)$$

By considering all face vectors, we can obtain a matrix size of $N^2 * M$, $A = [\ a_1\ a_2\ a_3 \ldots a_m\ ]$. This allows us to find the covariance matrix by performing A by A^T multiplication and because A has $N^2 * M$ dimensions, thus A^T has $M * N^2$ dimensions. Therefore, multiplying $N^2 * N^2$ results in N^2 of N^2 range that is computationally not useful to calculate. Hence, computing the covariance matrix is by multiplying A by A^T which results in $M * M$ matrix with M (presuming $M << N^2$) eigenvectors of size *M*. This results to:

$$Covariance = A^T A \quad (7)$$

where A is formed by the different vectors, i.e., $A = [\Phi_1,$

$\Phi_2 \ldots \Phi_M$]. The above covariance (C) formula helps to calculate eigenvalues and eigenvectors. In practice, the dimension of C is $N * M$. Considering the eigenvectors v_i such that

$$A^T A v_i = \lambda_i c \tag{8}$$

Then, remultiplying both sides of (8) by A we get

$$A^T A v_i = \lambda_i c \tag{9}$$

where, it can be noted that Av_i represents eigenvectors with λ_i equivalent to μ_i which represents the $C = AA^T$ and $\mu_i = Av_i$ eigenvalues. Therefore, this analysis concludes that C' and C eigenvalues are the same and the relation on their eigenvectors is $\mu_i = Av_i$. Therefore, constructing $M * M$ the M eigenvectors, v_i, of covariance matrix provides M the biggest eigenvalues of C'. Now taking the normalized trained faces x_i and to characterize every single face vector in a direct combination of the most exceptional K eigenvectors (where K < M) results to

$$X_i - \Psi = \sum_{j=1}^{K} w_j \mu_j \tag{10}$$

where, μ_j are called the eigenfaces. Fig. 5 depicts eigenfaces.

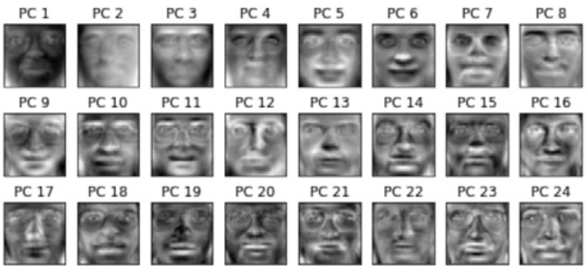

Fig. 5 Eigenfaces with Highest Eigenvalues [16]

Given a new face (Γ), preprocessing is performed to make sure the face is positioned in the center of the image and that it has identical dimensions as the face being trained. Therefore, a conversion to a set of eigenvectors is by subtracting the face from the average face $\boldsymbol{\Psi}$, as in:

$$\Phi = \Gamma - \Psi \tag{11}$$

Thus, the normalized vector is projected onto eigenspace to get the direct combination of eigenfaces simply by

$$\Phi = \sum_{j=1}^{K} w_j \mu_j \tag{12}$$

From this projection the vector of the coefficient generates weights to form a feature vector,

$$\Omega^T = [\, w_1 \; w_2 \, w_3 \; \ldots \; w_m \,] \tag{13}$$

This feature vector details the involvement of a single eigenface representing an input image and considers the eigenfaces a foundation place designated to face images. Thus, the class of a face Ω_k is computed through assessing a location of the results associated with the eigenface depiction across fewer individual face images. Categorization is achieved by subtracting a feature vector from the trained face image to get the minimum distance between the training and testing vectors. This is the Euclidean Distance among an input face image and faces classes. The aim here is to get the class k of the face that minimizes the Euclidean Distance, as in:

$$\varepsilon_k = \min_k \| (\Omega - \Omega_k) \| \tag{14}$$

where Ω_k is a vector describing the k^{th} faces class. If ε_k is lower than the forbearance level T_k, then it is recognized with k face from the training face image, otherwise, the face is not matched with any faces in the training set. Though this algorithm is computationally inexpensive it is sensitive to illumination, and it requires a frontal view of the face to work effectively.

G. Classification Using SVC

Support Vector Classification (SVC) class can perform binary and multi-class classification on a dataset. It is here preferred with Radial Basis Function (RBF) kernel, thanks to its effectiveness in high dimensional spaces [14]. When training a SVM with this kernel two factors must be considered: *C* and gamma parameters. *C* trades off all misclassifications of training face images against maximization of the decision surface. The aim is to get a high *C* to ensure all training faces are classified correctly. We chose $\mathrm{C} = 1000$ as this value showed a significant improvement. However, gamma (γ) verifies the effect of a single training that both the face image and bigger γ have, and the distance determines the effect of other face images. This parameter value was given as gamma = 0.001 as it also improved our classification.

III. Training and Simulation of CNN

From the ORL database of 40 classes, a CNN was implemented. First, the eigenfaces and FE are performed by computing the best values in the ORL dataset. Fig. 5 Section II of this paper showcases eigenfaces obtained during training. Fig. 6 depicts the cumulative explained variance ratio. Then we cluster these feature projection vectors using K-Means clustering and feed them as inputs in the CNN. Figs. 7 and 8 depict the schematic diagram of the proposed CNN training phase and classification simulations.

When FR is considered for a new face image, the feature projection vector of this new face is calculated from the eigenfaces, classified with SVC, clustered using the K-Means algorithm to isolate small parts in n groups of equivalent variation and choosing centroids that minimize a criterion known as inertia and then this facial representation obtains face signifiers. The face descriptors (identifiers) are then supplied to CNNs, thereafter, modelled with this network, to compare the network outputs. Decision-making is based on the minimum and maximum outputs. At the highest output, this new face is chosen to be the part of the class of an individual that has the highest output.

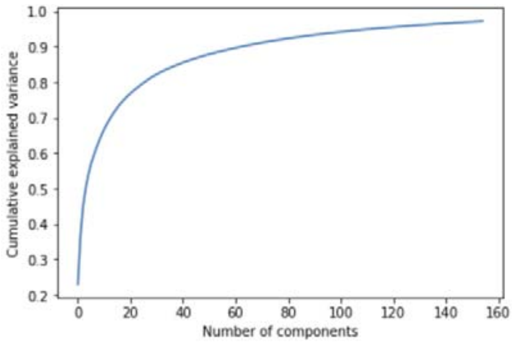

Fig. 6 Cumulative explained variance

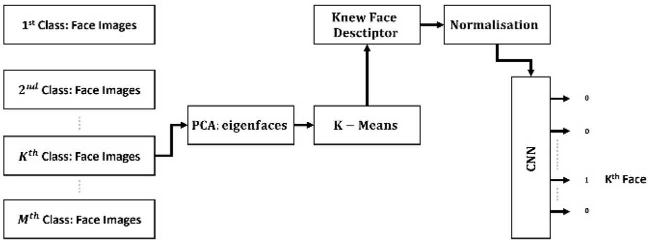

Fig. 7 CNN training phase

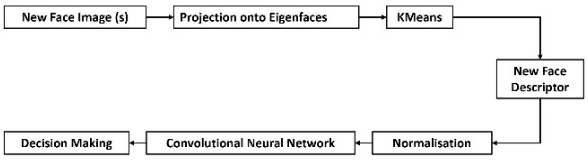

Fig. 8 CNN Classification Simulations

IV. Empirical Findings

To examine the proposed algorithm, the ORL_faces dataset was used. It comprises 10 different face images per class for every 40 distinct persons. These images are in greyscale format and were captured in different lighting conditions and occasions. Per class, one person is represented in different facial expressions such as an open or close mouth, with or without glasses, and with different head orientations [16]. These images are used for training and testing. Fig. 3 shows among the views some of these faces.

Splitting the dataset to a test size of 20% and 80% training improved the accuracy and our proposed methodology obtained 99% in 90 epochs. Figs. 9 and 10 display the graphical representations of the model accuracy and loss after training. Table I depicts the confusion matrix.

TABLE I
Confusion Matrix

Predicted Real	Numerical Form		
	Class_1	*Class_2*	*Class_3*
Class_1	3	0	0
Class_2	0	1	0
Class_3	0	0	2

Table II shows classification report where the f1-score accuracy is 99%.

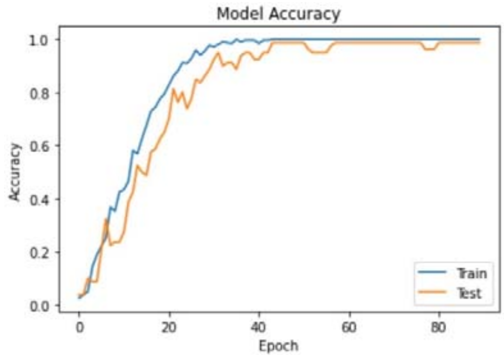

Fig. 9 Model Accuracy of 99%

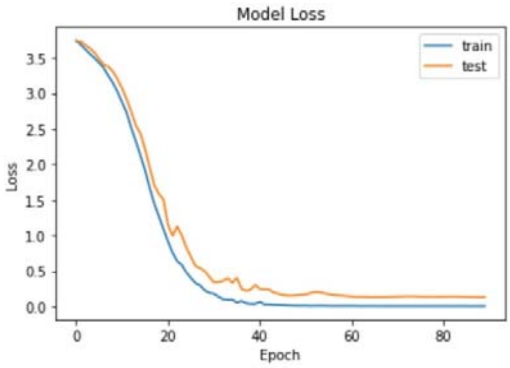

Fig. 10 Model Loss of 1%

TABLE II
F1-Score of 99%

Scoring Metrics	Classification Report			
	Precision	*Recall*	*F1-Score*	*Support*
Accuracy			0.99	80
Macro avg	0.99	1.00	0.99	80
Weighted avg	0.99	0.99	0.99	80

F1-score is a weighted average of the precision and recall whereby when it performs well it reaches a value of one and in poor performance, it scores a zero. Thus, our model's accuracy at its best is 0.99 which is close to 1.

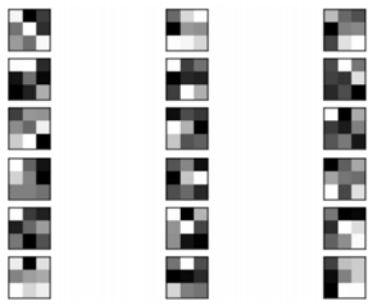

Fig. 13 A Filter of the first hidden layer.

Fig. 13 depicts the first 6 filters out of 64 filters of the first hidden convolutional layer of the proposed model with 18 face images. A single row represents one filter and a single column represents one channel. Inhibitory weights are the dark squares and excitatory weights are the light squares. Excitatory inputs are often positively weighted and valued while inhibitory inputs are negatively weighted and valued. Each neuron has a fixed threshold for firing, and this can be achieved by an excitatory input. To be able to view the feature maps of the input face image we visualize what features of the input are detected and preserved in the feature maps.

Fig. 14 depicts the feature maps in the first convolutional layer by showing different versions of the face image with different features highlighted. Some of these highlights can be seen as a focus on lines, and the background of the foreground.

Table III represents the obtained compared outcomes against other face algorithms such as PCA+SVC, SVM, Naive Bayes (NB), kNN, and the proposed CNN algorithm.

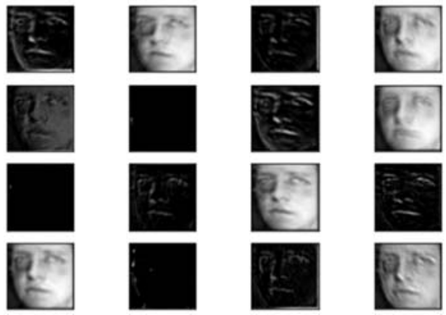

Fig. 14 First convolutional layer feature maps

TABLE III
PERFORMANCE ANALYSIS FOR THE PROPOSED METHODOLOGY

Test Results							
Database	Training Images %	Testing Images %	SVM	NB	KNN Performance %	PCA + SVC Performance %	CNN Performance %
ORL							
	50	50	94	48	61	93	91.5
	60	40	94	63	74	94	94
	70	30			76	96	91
	80	20	96	82	84	97	99

V. CONCLUSION

In this paper, the improved deep learning neural network (IDLNN) FR system is suggested and implemented in Python. It is based on PCA preprocessing, K-Means clustering followed by CNN. The FE vectors acquired from the PCA technique are utilized as input vectors in K-Means clustering and then normalized in preparation to train and test our CNN model. Our proposed method performed better than the PCA+SVC, SVM, KNN and NB and obtained the best accuracy of 99% at 90 epochs. The results suggest that CNN surpasses the abovementioned algorithms for the ORL database and that there is a room of improvement. For future work, it would be suggested that different databases are tested using this approach and that a live FR system is used to observe how accurate our classification is with different ethnicities.

REFERENCES

[1] S.V.A.V.Prasad & Shilpi Singha, 2018, ´ techniques and challenges of face recognition: a critical review ´, Procedia Computer Science, no.143, pp.536–543.

[2] Bhaskar, B., Anushree, P.S., Shree, S.D. and Prashanth, K.V.M. (2015). Quantitative Analysis of Kernel Principal Components and Kernel Fishers Based Face Recognition Algorithms Using Hybrid Gaborlets. Procedia Computer Science, vol 58, pp.342–347.

[3] Vahid Mirjalili and Sebastian Raschka. Python Machine Learning. Packt Publishing Ltd, UK Birmingham, 2017.

[4] Jarrett, K., Kavukcuoglu, K., Ranzato, M., and LeCun,Y. What is the best multi-stage architecture for ob-ject recognition? In Proc. International Conference on Computer Vision (ICCV'09). IEEE, 2009.

[5] Vinod Nair and Geoffrey E. Hinton. Rectified Linear Units Improve Restricted Boltzmann Machines, Department of Computer Science, University of Toronto, Toronto, ON M5S 2G4, Canada, 2010.

[6] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. and Bengio, Y. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*,15 (56):1929-1958.

[7] Long, J., Shelhamer, E. and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431-3440.

[8] Viola Paul and Michael J. Jones. Robust real-time face detection. International journal of computer vision 57, no. 2 (2004): 137-154, 2004

[9] Ren, S., He, K., Girshick, R. and Sun, J. (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Advances in Neural Information Processing Systems*, vol 28, pp. 91-99, 2015.

[10] Prateek Joshi, *Artificial Intelligence with Python,* Packt Publishing Ltd, Birmingham, UK, 2017, p. 408.

[11] Sebastian Raschka & Vahid Mirjalili, *Python Machine Learning,* Packt Publishing Ltd, Birmingham, UK, 2017, p. 12.

[12] Richard Ernest Bellman, *Dynamic Programming,* Princeton University Press, Rand Corporation, 1957, p. ix.

[13] Alaa Eleyan and Hasan Demirel. Face Recognition System Based on PCA and Feedforward Neural Networks, Department of Electrical Engineering and Electronic, Eastern Mediterranean University, Gazimagusa, North Cyprus, Mersin 10, Turkey, 2014.

[14] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. A Practical Guide to Support Vector Classification, Department of Computer Science, National Taiwan University, Taipei 106, Taiwan, 2016, p. 4.

[15] Zhao, F., Li, J., Zhang, L., Li, Z. and Na, S.-G. (2020). Multi-view face recognition using deep neural networks. Future Generation Computer Systems, vol 111, pp.375–380.

[16] E. I. Abbas, M. E. Safi and K. S. Rijab, 2017, ´Face recognition rate using different classifier methods based on PCA`, International Conference on Current Research in Computer Science and Information Technology (ICCIT), 2017, pp. 37-40