International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V71I8P234

Volume 71 Issue 8, 388-409, August 2023

Original Article

Deep Learning Approach for Face Recognition Based on Multi-Layers CNN&SVM

Abu Sanusi Darma[1,3], Fatma Susilawati Mohamad[1], Oladapo Ayodeji Diekola[2], Ibrahim Mohammed Sulaiman[4]

[1]*Faculty Informatics and Computing, University Sultan Zainal Abidin Terengganu, Malaysia.*
[2]*Department of Computer Information Systems, University of Houston Victoria, USA.*
[3]*Department of Computer Sciences & Information Technology, Al-Qalam University, Katsina, Nigeria.*
[4]*Institute of Strategic Industrial Decision Modeling (ISIDM), School of Quantitative Sciences, Universiti Utara Malaysia.*

[1,3]*Corresponding Author : darmasanusiabu@yahoo.com*

Received: 16 April 2023 Revised: 13 July 2023 Accepted: 21 July 2023 Published: 15 August 2023

Abstract *- The inspiration behind the huge attention given to face recognition systems by the research community and computer vision specialists is the need to enhance face recognition systems' effectiveness, accuracy rate, and speed. The complexity of recognizing the human face by machines due to different variations in poses, illumination, age, facial expression, occlusion, personal appearance, and different cosmetic effects makes face recognition more challenging. However, this makes it difficult to implement a robust computational system. The study's main goal is to enhance the current deep learning approaches for face recognition applications using an enhanced and efficient hybrid deep learning method that involves multi-layer CNN and SVM. The model is encompassed with a newly developed middle block convolutional regularization algorithm (MBCRA) and a pre-activation batch normalization method for computational stability and convergence speed. The combination of both CNN and SVM enables the system to obtain more significant face features from the images of the proposed AS_Darmaset. The database has six classes of images. Each class contains face images with specific variation problems. The experimental results demonstrate that the multi-layer CNN+SVM has a 99.87% accuracy, and the comparative analysis shows that the proposed model is more resilient for face image classification under unconstrained settings than the most developed deep learning model for face recognition.*

Keywords *- Deep learning, Face recognition, Convolutional Neural Network, and Support Vector Machine.*

1. Introduction

Face recognition systems (FR) have progressed to become a frequently used form of biometric system, one of the most important and difficult areas of computer vision [1]. It is now the most active field of research demand, and it is an important technology in the fields of security, business applications, law enforcement, and many more. With more emphasis on its utilization for biometric analysis, human computing interaction (HCI), surveillance systems, and content-based coding of images [2].

Despite the availability of numerous presence biometric verification systems such as hand geometrics, iris scans, retinal scans, and fingerprints, human facial recognition applications will continue to be the most authenticating system due to the numerous features they possess. Features such as low cost, absence of physical contact between the user and the system, and user acceptance. The system will continue to have huge significance in the area of security that provides intelligence services since the concern about proper security systems has reached its maximum point [3].

Face recognition has touched many impotent areas of biometric applications, such as bank authentication systems, law enforcement identification systems, security building access control, personal identification systems, and digital security systems, among many more. The face of a person conveys information concerning the person's identity and state of emotion [3]. According to Bennamong [4], face recognition encompasses various biometric surveillance security systems such as (e.g., border control, suspect tracking, and terrorist identification), security systems (e.g., system login, internet access, and file encryption), medical treatment (e.g., facial surgery, and maxillofacial rehabilitation), and entertainment (e.g., human-computer interaction and video games). It is widely considered an important technology in the area of digital verification and identification tasks to verify and identify a human face identity based on the statistical features or geometric representation of the person’s face image [1].

When compared with other biometric systems such as fingerprints, iris image scans, and retinal scanning, the face recognition system is a large and socially accepted

identification system because its acquisition is natural, nonintrusive, and physically contact-free. Two different modalities are commonly defined for facial recognition: a 2D image database (which includes color and grayscale images) and a 3D data set consisting of depth images, point clouds, and meshes [5]. Many CNN models used in face recognition systems require a lot of training and testing data. This increased network performance and led to a high recognition rate. Although large-scale databases, such as the ImageNet, Large Scale Visual Recognition Challenge (ILSVRC) [6] and [7], have been involved in the domain of a massive CNN-based leap forward occurring in the current computer vision research community, such databases were discovered to be very absent from the domain of face recognition. The most recent advancements in the field have come from the well-known internet behemoths Facebook and Google [8,11].

The current powerful facial recognition database [8] has been trained with almost 200 million human face images of eight million people's identities. The size of this face database is about three orders of magnitude higher than other publicly available human face databases [9].

A deep learning technique is introduced in this study. To extract and classify face characteristics, the method uses a hybrid model that combines a convolutional neural network with a support vector machine (CNN+SVM). Because of its classification capacity, the model uses the SVM as the final classifier once the CNN has finished retrieving features. The Support Vector Machine will extract more features and then use the collected features to classify the data. This is achieved by utilizing the face characteristics collected by CNN.

The study made use of two freely available internet image databases: Label Faces in the Wild and Caltech-101-ObjectCategories. The study chose some face images from these two databases to construct a strong dataset suited for training and testing the proposed model. AS_Darmaset was the name of the database. This collection contains 5280 human face images of 135 people.

2. The Following are the Contributions of this Research Paper

This paper investigates the impacts of regularization techniques by adding Batch Normalization Layers, Dropout layers, and data augmentation techniques to improve CNN training, performance and to prevent network overfitting.

This is the first study to add MBCRA to design a multi-layer CNN+SVM with training and testing accuracies of 100% and 99.87%, respectively**.**

The study demonstrates that multi-layer CNN, when combined with SVM on a large number of training and testing datasets, is more robust than the basic standard CNN of a few layers of architectural models, which may effectively operate well with a small number of datasets and fewer network layers.

3. Face Recognition Approaches

There are two types of facial recognition approaches: machine learning and deep learning approaches. Despite the fact that deep learning is a subclass of machine learning, the two have similar traits and perform similar functions. They all have varying levels of performance, accuracy, and speed [9]. As a result, this study examines face recognition techniques from these two distinct perspectives: 1) machine learning-based techniques and 2) deep learning-based approaches; machine learning is the only technology entirely based on feature extraction techniques.

Feature extraction is the application of extracting algorithms on computer images to reduce duplication and unnecessary images. The primary importance of this strategy is to lower the complexity of space and speed up machine training time to achieve face dimension reduction operations [10]. There are two variants of this technique: 1) model-based, which makes use of geometric face features such as the lips, eyes, brows, and cheeks, as well as the geometric connections between them. 2) appearance-based, in which holistic texture elements are given to either the complete face or specific areas of the facial picture [11].

These approaches enable image dimension reduction to a minimal face dimension while retaining substantial facial features [12]. On the other hand, deep learning is primarily a self-controlling, self-training system that utilizes the already present data to train algorithms further to find complex patterns. It can be regarded as a machine learning approach that attempts to extract high-level and hierarchical characteristics from a given row of data, such as images, audio, and videos [13].

Deep learning approaches include Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), Recurrent Neural Networks (RNN), Multilayer Perceptron Feedforward Neural Networks (MPFFNN), and Stacked Auto-Encoder (SA). These are the most frequently used deep learning models [14]. The purpose of these techniques is to limit the impact of the primary elements influencing the face recognition system and to construct a strong face recognition system [5].

To cope with differences in facial expression, cosmetic impacts, occlusion, age, and position variation, this research study proposed face recognition systems based on multi-layer convolutional neural networks (CNN) and Support Vector Machine (SVM) All of the above are unconstraint conditions that are affecting the face recognition system.

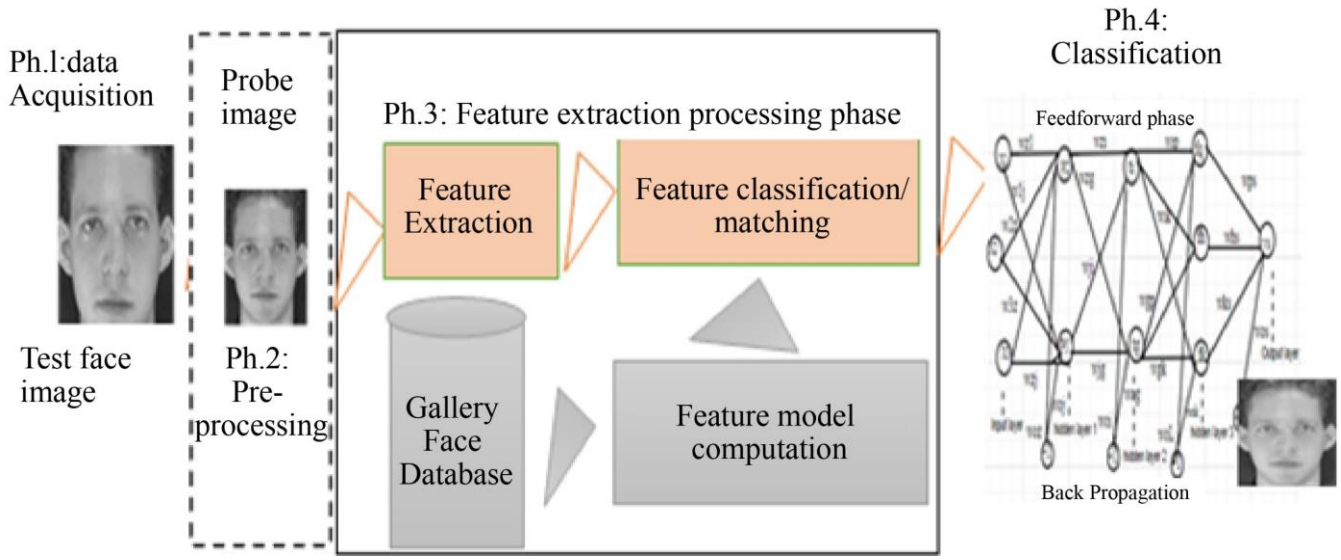

Fig. 1 Generic structure of face recognition

4. Face Recognition System Pipeline

Face recognition is an easy task for the human brain, but it is incredibly tough for a computer to perform artificially due to many similarities, despite age and gender disparities. Variations in image quality, facial emotions, facial furniture, backdrop, and lighting circumstances exacerbate the situation [16]. Figure 1 displays a general depiction of a face recognition system's pipeline. The pipeline is divided into four phases [17].

4.1. Data Acquisition (input) Phase

This phase deals with data collection techniques. It consists of all the available techniques employed to collect a large amount of training and testing data. The process is solely relayed on different kinds of computer devices for image input, such as scanners, digital cameras, digitizers, etc. At this stage, a face image passes into the system through an image-stored database or from any computer device for image classification [50].

4.2. Pre-Processing Stage

In every face recognition, image pre-processing is important to ensure that every face image has obtained all sorts of standard information before the identification and verification process begins. It is necessary to ensure aligned images of equal size, position, color format, good resolution, and a unique image format [18]. Pre-processing is the process of removing noise and unwanted elements from an image [49].

4.3. Feature Extraction Stage

The feature extraction process deals with identifying and extracting unique facial features like noses, mouths, eyes, and other fiducial marks on the face images, such as wrinkles, moles, freckles, dark points, etc. Facial feature extraction can be achieved using a robust transformation method [52]. The image dimension can be resized to a smaller dimension by retaining significant face features [19].

4.4. Classification Stage

This is the last stage of the face recognition system pipeline, and it includes strong classifiers like multi-layer feed-forward backpropagation neural networks, Support Vector Machines (SVM), Euclidean distance classifiers, CNN and so on. The unknown face image is compared to images from the database during the classification phase [20]. The classification method requires a powerful algorithm output or the characteristic distance between the input feature vector and the face dataset gallery reference vectors [21].

5. Review of Related Work

Due to numerous research studies on face recognition systems, this research paper tried to investigate various architectures for face recognition systems. To develop and implement a robust face recognition system, many researchers in the computer vision community tend to think of new approaches based on deep learning techniques such as Convolutional Neural Networks (CNN), Stock De-Noising Auto-Encoder (SDAE), Deep Belief Neural Networks (DBN), etc. However, much of the current face recognition research has employed a different variant of CNN architecture. This paper reviewed some of these exhibit choices to screen what is significant from relevant details since CNN has been used as a powerful biometric system for solving numerous verification and identification problems. The deep learning model has been inspired by the biology of human vision [22].

Gonzalez et al. [23] presented the deep CNN in 2012 as a method to analyze the ImageNet database and get better results. Several CNN designs have been presented in the literature [24] and have been used to achieve better results

than the current state of the art. In their research, Ahmad et al. [2] created a MATLAB-based CNN for face recognition. To minimize network training time, the proposed Convolutional Neural Network Method can receive new inputs by training the last two layers of the model. The experiment used face images of 40 people from the AT & T database and 10 people from the JAFTE database, and the result was 100 percent accuracy in less than a minute of training time.

Y. Li et al. [15] proposed a CNN-SVM hybrid technique to recognize human faces. The CNN is used for feature extraction, while the Support Vector Machine (SVM) will detect face images more effectively using the input of facial features extracted by the Convolutional Neural Network (CNN). The experiment results show that the model is more efficient, with a greater recognition rate and a shorter training time. Parkhi et al. [6] developed a facial recognition system employing a very deep CNN and corresponding training system to obtain higher face recognition accuracy when compared to the current trend on the public benchmark.

This study evaluates how a large-scale database (of 2.6 million images) spanning over 2.6 thousand individuals could be developed by semi-automatic annotation with a human in the loop. Mohamed [26] aimed to increase the accuracy of a 2D face recognition system by learning discriminative features using a CNN of 15 layers. The network is trained using the stochastic gradient descent technique. Their research suggests that the Face96 database has a 99.6% accuracy rate.

Benkaddour et al. [24] proposed a hybrid deep learning-based face recognition and identification method. The approach employs a convolutional neural network (CNN), a support vector machine (SVM), and principal component analysis. The CNN was utilized for feature extraction, while the SVM was used as a classifier. They used the Principal Component Analysis (PCA) approach to reduce the dimensionality of face features.

The findings of their experiment reveal that their proposed technique has resulted in a significant improvement in model recognition accuracy. Najm et al. [27] proposed integrating and exploiting CNN for human face recognition and categorization in various applications. Their paper presents a deep learning-based CNN approach with fuzzy logic to get a higher degree of exactness in the facial grin. Using this technology, the predictive characteristic of the human face may be used for a criminal investigation of the social analytics-based application.

6. Materials and Methods

6.1. Methods

This research aims to achieve better network performances and human face recognition with a higher recognition rate, without system overfitting and with less computational load. Image registration, powerful feature extraction, and classification models are the key points to achieve the above aims. To achieve the aims, the study employed the use of a hybrid deep learning method that involved multilayer CNN+SVM.

In addition, a new middle block convolution regularization algorithm (MBCRA) was developed to enhance stabilized network training and processing speed [28]. The multilayer CNN is for feature extraction and classification. The SVM in this model is responsible for further feature extraction and final classification, utilizing the face characteristics extracted by CNN. This hybrid system is capable of extracting more features than the standard CNN [20].

The SVM has several advantages in addressing high-dimensional pattern recognition and nonlinear classification applications. On the other hand, the CNN model can receive images as direct input, manage them for rotation, scaling, image distortion, and translation, and can automatically extract useful face characteristics [29]. The research also adopted the mini-batch stochastic gradient descent training technique to train the proposed models. A batch size of 400 was used for the training of the models. This means 400 samples from the training dataset will be used to estimate the error gradient before the network weights are updated.

The strength of the model is based on adding the new MBCRA to the model architecture and a step data augmentation technique after the normal pre-processing step to generate and build a suitable dataset for the training and testing of the proposed model. The database is derived from the two public benchmark databases, namely the Caltech 101_ObjectCategory database and the Label Faces in the Wild database.

6.2. System Model Design

Most of the very simple CNN structures developed by numerous researchers originate from LeNet-5, which contains a very simple topology of layers [2]. LeNet-5 encompasses seven (7) layers, all of which contain trainable parameters (weights), with an input size of 32x32 pixel images [30]. This research proposes to study and develop an enhanced deep learning approach for face recognition.

The research proposes to study the architectural design of a simple LeNet-5 convolutional neural network and to modify the hybrid CNN+SVM developed by [2] by adding more layers to their original 7-layer architecture. The structural topology of the proposed multi-layer CNN+SVM will have multiple layers of architecture. The improvement of the model structure is achieved by adding more layers and network features of MBCRA that ensure a high recognition rate while maintaining network training stability, performance, and processing speed.

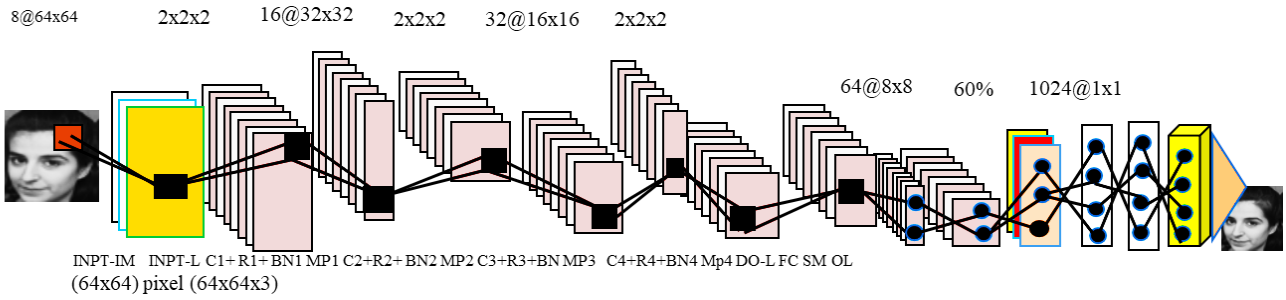

Fig 2. The structure of the proposed multilayers CNN

6.2.1. The Topology of the Proposed Multilayers CNN

The Architectural Designed of the proposed multilayer Convolutional neural network consists of a single input layer (1) labeled as INPT-L in Figure 2 above, four (5) convolutional layers labeled as C1, C2, C3, C4, and C5, respectively, in the diagram. The structure has four (4) Rectifiers Linear Unit (ReLu) layer label as R1 to R4, four (5) Batch normalization layers label as BN1 to BN5, four (4) Max pooling layers MP1 to MP4, one (3) Dropout layer label as DO-L, two (2) fully connected layers, and SVM classification layer. Each layer in this topology is a connected layer of linear mapping of a different number of face image data [31]. Figure 2 shows the pictorial representation of the Proposed Multilayer CNN.

Input Layer (INPT-L)

This is the first layer and serves as the system's gateway that enables the human face image to pass into the system. The entire sample images that could pass through the input layer were scaled to the scaler vector in which the height and the width are in pixel sizes, preprocessed with cropping, flipping, and some of the images were augmented using numerous augmentation techniques to enable the system to produce an effective, robust, and accurate recognition rate. The output of the input layer is passed to the convolutional layer for feature map extraction, which will then be passed to the subsequent layers as input from the convolutional layer. The feature map of every layer is the input set of the next layer [26].

Convolutional Layer (C1 to C5)

In the design principle of the proposed deep learning model, there are five convolutional layers. The convolutional layers are labelled as C1 for the first convolution layer, C2 for the second, C3 for the third convolutional layer, and one of the layers that make up the MBCRA in the middle convolutional unit, C4 for the fourth convolutional layer, and C5 for the fifth convolutional layer, respectively. Every input data point in the filter is joined to a [3x3] depth convolutional kernel, a local receptive field in the input layer of the size of [64x64] C1. The spatial size of the output volume in the proposed model is a function of the input volume size of $W = 64$ the receptive field size of the layer $F = 3$, the stride with which they are applied, and the amount of padding on the border edge. These can be represented as ([64x64x3]), S=1, P=1, F=3.

To obtain the number of neurons and their connected weights on this layer, the study uses the formula.

$$\frac{W-f+2P(1)}{S}+1$$

This gives $\left(64-3+\frac{2(1)}{1}\right)+1=64$ the total number of neurons in this convolutional layer[64x648x8] =32,768. Each of these neurons has a weight of ([3x3x3]x1)x8) = 224. The filter [3x3] size is the same as above $F = 3$ and is the number of depth channels from the input volume. In summary, the first convolutional layer of the proposed model has 32,768 neurons, each connected to 224 weights. With the first max-pooling operation, the second convolution (C2) layer in the proposed model has ([32x32x16])=16384 neurons. Each is linked to the weights (3x3x8) x 1 x 16) = 1168.The third convolutional layer, C3, has ([16x16x16]) = 4096. Each neuron is connected to ([3x3x16] x1) x16) = 2304 weights and biases of 16, respectively. The C4 layer has (16x16x32) = 8192 neurons and has (3x3x16) x 1) x 32) = 4608 connected weights for each neuron. Lastly, the fifth convolutional layer, C5, has ([8x8x64]) = 4096 neurons and (3x3x32) x 1) x 64) = 18432 connected weights.

It can be observed that the neurons are reducing. drastically due to the number of max-pooling operations that occur at the end of every convolutional operation, where the input volume is reduced to half of its size. The logical structure and the number of hyper-parameters of the the proposed model enhances the recognition rate and weight sharing. As a result, the model is more similar to a biological brain network. Weight sharing reduces the the complexity of the face recognition system and the number of parameters that must be calculated [2]. Figure 2 is the model diagram. The convolutional formula can be described as follows:

$$y_j^l = f\left[\sum_{l\epsilon M} x_{ij}^{l-1} * K_{ij}^l + b_j^i\right] \quad (1)$$

Where is y_j^i the jth output map at the layer l,, where x_{ij}^{l-1} is the $i\ th$ input feature mapping at the layer $l-1$, and M is the set of feature maps in the $l-1$.. K_{ij}^l is the depth

convolution kernel between the ith input map at the layer $l-1$ and the jth output map at the layer l. b_j^i, which is the output feature map at the layer l.

Rectified Linear Unit (ReLU)

The ReLU function $f(x) \longrightarrow (0, x)$ describes neural signal activation. A typical convolutional layer of CNN is an affine map $R^m \longrightarrow R^n$ that needs to involve the non-linearity function $R \longrightarrow R$ ReLu function $x \longrightarrow (0, x)$ in its convolutional process, where every R^n layer transforms one volume of activation to another $R \longrightarrow R$ layer by passing the weighted sum of its inputs through this differentiable activation function.

During the convolutional operation, the convolutional layers of the proposed multilayer CNN applied an elementwise activation function $x \longrightarrow \max(0, x)$ thresholding at zero. This leaves the size of the volume unchanged. The ReLu function helps the CNN to achieve its true potential by passing the result of the convolutional operation through the ReLu function to optimize the Neural Network weight. The ReLu then passes the final feature maps to the Batch Normalization layer, which will do the normalization operation. The real function has $f(x) = \max(0, x)\ (x\ \sum(0 + \alpha))$,.

The gradient formula is as follows $I\{x > 0\}$.: As a result, the problem of gradient disposition in the backward propagation process is eased, and the parameters in the network's first layer are swiftly updated. The ReLU function established the threshold for low computational complexity.

$$\text{if } x < 0 \text{ then, if } x > 0, then\ f(x) = x \qquad (2)$$

Batch Normalization Layer (BN)

The batch normalization is applied to the network to normalized the convolutional operation to reduce memory utilization, decrease learning time, and prevent overfitting. This helps the network to output more stable predictions through regularization and speed up training by an order of magnitude. The normalization process is carried out within the scope of the non-linear activation layer of the function $f(x) \longrightarrow \max(0, x)$ by subtracting the mean of the batch activation and dividing the subtraction result by the standard deviation of the batch activation to normalize the data on the same scale.

This means that an affine transformation is applied to each unit $z^{[l](i)}$so that the mean and variance of the set $z^{[l](i)} \ldots z^{(m)} \in R$ are zero and one. Then to normalize the previous output value to the same scale value, an affine transformation $z^{n(i)} \longrightarrow \gamma * z_{nor}^{(i)} + \beta$ is applied to each unit, as shown in the formula below:

$$\mu = \frac{1}{m} \sum z^{[l](i)} \qquad (3)$$

$$\sigma^2 = \frac{1}{m} \sum(\text{z:} - \mu) \qquad (4)$$

$$z_{nor}^{(i)} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^{2+} \sum}} \qquad (5)$$

$$z^{\mu(i)} = (\gamma * z_{nor}^{(i)}) + \beta \qquad (6)$$

The output $z_{nor}^{(i)}$, of normalization, is obtained by subtracting the mean μ of the output layer l, and dividing it by the standard deviation while $\boldsymbol{z^{\mu(i)}}$ setting the mean μ and variance σ^2 to the new value $\gamma\ and + \beta$ denotes parameters that can be learned.

Max Pooling Layer

Pooling operation is done by max-pooling layer, which consists of 2x2 filter size applied to stride-down samples. Divides each depth slice in the input 2 along both width and height, discarding 75% the activation. In the architecture of the proposed model, Layers, MPL2, MPL3, and MPL4 are max-pooling layers commonly known as subsampling layers, whose number of feature maps is equal to the number of the maps of their previous convolutional layer. The neuron of the first convolutional layer is the feature map of the layer, which is 8, then the feature map of the output volume of the MPL1 is (32x32x8) and its feature maps is also 8.

The function of this layer is to reduce the spatial size of the representation by resizing the number of parameters and to reduce the computational load in the network [1]. In the max-pooling layers, each neuron in the feature map is linked to a local receptive field in 2x2 the kernel size of the preceding layer. These layers do the down-sampling operation alongside the spatial dimensions (width, height), ensuing in volume ([2x2]), since the study used a stride of 2.

The max-pooling decreases the dimensionality of the image by decreasing the number of pixels from the output of the preceding convolutions layer. The proposed system works by defining an end-by-end region as the corresponding filter for the max pooling operation, which has a 2x2 filter size and a stride of 2 max value; in this stage, the convolutional kernel stride over 2 pixels at a time halving the size of the output.

This helps reduce the max values and resolution of the given output of the convolutional layer. This minimized the number of parameters, which led to the reduction of computational load. Figure 3 (a) depicts the 4x4 feature maps, which are the output of the convolutional layer, whereas Figure 3 (b) displays the maximum values of the downsampling operation, which is the output of the max-pooling layer.

The pooling filter in the figure shows the down-samples of the volume spatially and independently in each depth slice of the input volume. The max-pooling operation pools an input

volume of size (4x4) with a filter size of 2x2, stride 2 into the value of (2x2 window) size.

This research paper adopted to use max-pooling with the maximum pooling formula:

$$y_{j\,(m,n)}^{(l)} = f\left(\sum_j x_j^{(i-l)} * w_{i.j}^{(i-l)} + b_j^{(i)}\right) \quad (7)$$

In the above pooling formula, the $m \geq 0, n \geq 0$ and $y_{j\,(m,n)}^{(l)}$ stand as the value of neurons $unit\ (m,n)$ that are members of jth output feature maps. $w_{i.j}^{(i-l)}$ Represent the weight of the i neurons in the j class of the i layer, $b_j^{(i)}$ represent the offset of the i class, $*$ represent the convolution operation; $x_j^{(i-l)}$ represent the output of the j neurons in $l-1$layer, i.e., the input data for the l layer the activation $f(\ldots$ represents the function In the above pooling formula the $m \geq 0, n \geq 0$ and $y_{j\,(m,n)}^{(l)}$ stand as the value of neurons $unit\ (m,n)$ that are members of jth output feature maps. $w_{i.j}^{(i-l)}$ Represent the weight of the i neurons in the j class of the i layer, $b_j^{(i)}$ represent the offset of the i class, $*$ of the model that has the nonlinear characteristic.

The output volume of the first Convolutional layer is (64x64x8), which is the input volume of the first max-pooling layer. The max-pooling will use the formula $(W - F + 2P)/(2 + 1)$ to calculate and reduce the number of neurons by taken $W = 64$, $F = 3$, this depth size remains unchanged $S = 2$, and $P = 0$. This gives us $[(64 - 3 + 2(0))/4 + 1] = 14$ now the model neurons are reduced to $(14x14x8) = 1568$; each is connected to $(2x2x3) = 12$ weight. This shows that the pooling layer

- Accepts a value of size (W1xH1xD1)
- Required two hyperparameters
    - Their spatial extend F
    - The stride S
- Produces a volume of size

$$W2 = \left(\frac{W1-F}{S} + 1\right) \quad (8)$$

$$H2 = \left(\frac{H1-F}{S} + 1\right) \quad (9)$$

$$D2 = D1$$

Introduces zero parameters since it calculates a fixed function of the input.

Single depth Slice

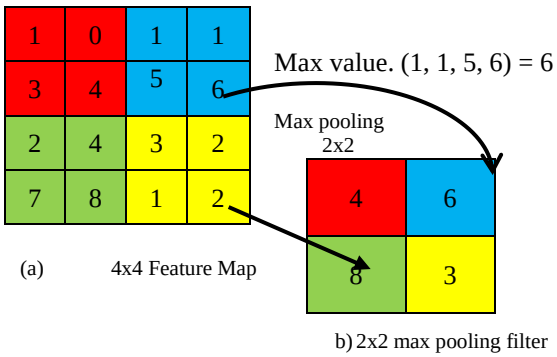

Fig 3. Max pooling operation on 4x4 convolutional output to feature map (2x2 windows)

Dropout Layer

A dropout is the weighted average of a model's estimated or expected result. It is one of the regularization approaches used to prevent the system from overfitting. During the training phase, the random selection of dropout nodes may overlook the hidden layer nodes [33]. There are three dropout layers in the structural arrangement of the proposed model.

The first dropout is a 45% dropout placed in the third convolutional unit, the second has a 60% dropout, which is placed at the end of the fifth convolutional unit, and the third dropout has a 69% dropout, which is placed in between the two connected layers. The proposed system has five convolution layers, five batch normalization layers, and four max-pooling layers. In all of the above-mentioned layers, the Rectifier linear unit (ReLU) is used as an activation function.

The kernel size of all convolutional layers is 3x3, and all convolutional layer inputs are padded with zero labels. The max-pooling layers employ 2x2 max pooling, and a stride of The dropout layer after pooling layer 4 is set with a probability of 0.60, which is a 60% dropout at random. This means that in every hiding layer of the network, the model is to drop out 60 of its neurons, picking the dropout neurons at random. The dropout regularized the network by adding noise to the output feature maps of each hiding layer to yield robustness to variation images so that neurons will not learn redundant details of input. This will improve the network performance and help prevent network overfitting.

Fully Connected Layers

The fully connected layer takes the results of the convolutional and pooling layers by flattening the result and reaching the classification decision. An image is composed of small details of features. The fully connected layers leverage these parts of the image features and utilize their various layers for analyzing each feature in isolation, thereby informing decisions about the image as a whole. In a fully connected layer, every input neuron is connected to every output by weight, which solves the purpose of doing actual

classification. Neurons in the fully connected layer have a full connection to all activations in the previous layers.

The sole difference between the fully connected and convolutional layers is that neurons in the convolutional layer are only linked to a local region in the input. So many neurons in the convolutional layer share the same parameters. However, the batch layer neurons continue to compute the dot product. Therefore their roles are not identical. The fully connected layer in the proposed system has that is at some input volume size that can consistently expressed as a Convolutional layer. In other words, the study set the filter size to be approximately the size of the input volume, so the output will just be a single depth column fit over the input volume. The fully connected layer allows the proposed model to sort out the expected prediction classes by flatting feature maps through an artificial neural network.

An artificial neural network consists of a flattening layer as an input layer, a fully connected layer, a softmax layer, and an output layer. In ANN, the completely connected layer is analogous to the hidden layer.

The study gets the predicted output facial expression and other classes in the output layer. So the study passes the whole feature vectors, which were obtained as part of the flatting operation, to this input layer of the fully connected ANN and then passed them to the fully connected layers, which will then combine the features into more attributes for the predicted classes.

The classes include human faces with makeup, human faces with pose variation, faces with facial expression, and faces with occlusion; the flattening layer of the proposed CNN has input neurons in the fully connected layer, and the output layer has the same attributes and output neurons representing the classification classes. Each of the six classes of neurons is defecting in one class of image category. The model is designed to predict each face image with at least a probability, which means it predicts wrongly given an error; this error is called the loss function.

The lost function of the predicted class is calculated and then backpropagated into the system to improve the prediction. It is to minimize the value of the lost function as well as optimize the network weights, which are the vector for most applications, particularly in deep learning [5], mini-batch gradient descent is the recommended gradient version for most applications. The number of samples given to the network in one training cycle results in one model parameter update [6]. Because of its short length, the mini-batch is commonly referred to as "batch size," and it is frequently tuned to an aspect of the computational architecture when the implementation is being executed, such as the power of two that fits the memory requirements of a GPU or CPU, such as 32, 64, 128, 256. We have introduced a new batch size that will work well with the proposed models. 400 for a mini-batch. This is effective when combined with the proposed multilayer CNN model [30]—the line of synapses. The fully connected layers operate in the form of feedforward and backpropagation operations.

SoftMax Layer

The Softmax layer is a probabilistic classifier that extends logical stagnation to multi-class situations with multiple discrete outcomes [2]. Image classification of 6 different classes is an example of a multi-class problem. In this research, six classes of different face image variations were proposed, including facial expression, facial makeup, occolution, oldage, pose, and young age. Softmax computes each attribute and weight using a stochastic gradient descent method [2].

It is a kind of soft continuation version of the function. The softmax output unit employs nonlocal nonlinearity; Softmax $(xTwj)^{T}xTwj$. Softmax is a smooth approximation of classification function, taking an input vector $\{x\}$ and weight vector $\{wi\}$, the predictive probability of $y = 1$

$$yi = \frac{e^{xTwic+bc}}{\sum_{k}^{k} e^{xTwi+bi}} \tag{10}$$

Where yi is the predicted class, x is the vector coming from the previous layer, w and b are respectively the weight and the bias associated with each neuron of the output layer [3]. This shows that the output yi is equal to the e^{xTwi} divided by the summation of that same quantity for all of the different Neurons in the softmax group. The yi represent the probability distribution just by using the softmax equation. The softmax equation has a simple derivative:

$$\frac{dyi}{dX^{Twi}} = yi(1 - yi) \tag{11}$$

This shows that yi changes when changing the X^{Twi}, but the X^{Twi} depends on the other values while yi depending on X^{Twi}.

7. CNN Training Hyperparameters

For a better and more robust network training process, several training option hyperparameters must be considered. To achieve better network performance, these hyperparameters must be appropriately defined. The hyperparameters defined for the proposed multilayer CNN training are:

7.1. LearnRate

The learning rate is the hyperparameter that governs how much the model changes in response to the predicted error at each update of the model weight. It is a customizable hyperparameter used in neural network training with a tiny positive value, often in the range of 0.0 to 1.0 [4]. The study utilized a learning rate of 0.0100 to train the proposed model.

After experimenting with the different number of learning (LR) parameters, the study determined that this is the best learning rate for the proposed model.

7.2. Mini-batch Size

For most applications, particularly in deep learning [5], mini-batch gradient descent is the recommended gradient version for most applications. The number of samples given to the network in one training cycle results in one model parameter update [6].

Because of its short length, the mini-batch is commonly referred to as "batch size," and it is frequently tuned to an aspect of the computational architecture when the implementation is being executed, such as the power of two that fits the memory requirements of a GPU or CPU, such as 32, 64, 128, 256. We have introduced a new batch size of 400 samples simultaneously, which works well on the proposed hybrid model [30].

7.3. Epoch

The number of epochs signifies the number of times the complete dataset has been transmitted forward and back through the neural network at one epoch.

An epoch occurs when each image has been seen at least once throughout the training. While each iteration is assigned a number, the overall number of forwarding and backward passes is defined by the batch size, the number of epochs, and the number of training images. It is calculated as follows:

$$Itration = \frac{Epoch\ x\ Training\ image}{batch\ size} \qquad (12)$$

The Max Epoch for the model is 90.

8. Datasets

The data acquisition process concerns the methods used to obtain a large number of training and testing datasets for the proposed model. In this paper, the study proposed collecting a sufficient number of datasets from two publicly available databases, namely the Caltech-101-Object-Category database and the Label Face in the Wild (LFW) Database (shown in Figure 4). Caltech's-101-Object-Category database has 450 facial photos of 27 different persons. The photographs are 325 × 495 pixels in Jpeg format, with varying expressions, backgrounds, and lighting, but this study advocated cropping and reducing each face image to 64x64 pixels.

Fig. 4 Shows the sample face images from the Caltech 101_ObjectCatogary database

The LFW database, on the other hand, comprises 13,233 target face images of 5749 different people. There are 1680 people in the database who have two or more photos. The remaining 4069 people have only one image in the database [38]. Figure 5 illustrates the image samples of LFW.

The images are available in JPEG format with a 250 x 250 pixels resolution. The majority of the images are in color, with only a few in grayscale. In this study, the study recommended using 4,200 selected face images from this database. For the training and testing of the proposed models, all of these images are in the color scheme, resized to 64x64 pixels, and in their original jpg format.

Fig. 5 Shows the sample face images from label faces in the wild in resize scale of (64x64) pixels

The study proposed utilizing 5280 human face images of 132 different individuals. Each individual has 40 distinct facial face images. The study employed dynamic data augmentation and preprocessing approaches to build multiple synthetic face images from each individual's face images.

Using the Caltech-101-objects-categories database, the study developed 1,080 face images of 27 people and 4,200 images of 105 people using the Label Faces in the Wild (LFW). This makes the study to have used a total of 5,280 face images. For both training and testing of the proposed model, the 5280 images were split into 5280/100 x 70 = 3,696, which is 70%, and 5280/100 x 30 = 1,584, which is 30%. The first 70% is for training, while the last 30% is for testing.

From Table 1, the proposed AS_Darmaset database has the largest collection of 5280 face images outside the LFW.

Table 1. Database comparison

Datasets	People	Images
Caltech 101_objects_Categories	27	450
Label Faces in the Wild	5,749	13,233
AS_Darmaset Proposed Dataset	132	5280
Face96 Dataset for 9 Layers CNN & SVM by [15]	200	2200
FERET Dataset for 15 Layers CNN by [26]	152	3040

9. Experiments

This section discusses the experimental results of four different deep-learning models for face recognition systems. The proposed hybrid multilayer CNN+SVM model, along with 7-layer CNN, 9-layer CNN+SVM, and 15-layer CNN models, are the four deep learning models. Experiments were conducted using the proposed AS_Darma set of 5280 face images of 135 individuals. This database was made by integrating some face images from two publicly accessible databases. The two databases are Caltech 101 Object Category and Label Faces in the Wild.

These experiments were conducted to investigate the benefits of employing a sophisticated deep learning algorithm to increase the performance of the face recognition system. The study compared the accuracy and loss errors generated by each of the four deep learning models. To aid in the selection of the best and most powerful model in terms of performance and processing speeds. Furthermore, the experiments investigated the ability of the regularization approach MBCRA used in the proposed model architecture to increase network training stability, speed and performance.

9.1. Experiment with Hybrid Model of Multi-Layers CNN+SVM

The study first used the proposed model for segmentation, feature extraction, and classification, all on a hybrid deep learning model [39]. MATLAB R2018b was used in developing and implementing the four deep-learning models. It was also employed to carry out the experiments.

Because it is the ideal programming language for engineering and artificial intelligence systems, the model's architecture is intended to run on the HP Elite Book 854w Mobile Workstation. An Intel Core i7 M620 powers the system @ 2.67GHz CPU and has 8.00GB of internal physical memory. Four experiments were carried out. The first experiment was on the proposed multilayer CNN+SVM, the second was on 15-layer CNN, the third was on 9-layer CNN with SVM, and the fourth was on 7-layer CNN.

The studies were conducted using the proposed AS_ Darma_set, which has six different classes of face image variants. These face-ace image variations are classified into six categories: cosmetic effects, facial expression, occlusion, faces of older age, pose variation, and faces of younger age. There are 880 face images in each of these classes. Each class has the face images of 22 persons, and each person has 40 64x64 pixel face images.

The proposed model's architecture was outfitted for this experiment with a newly constructed MBCRA, a regularization method involving a convolutional layer, batch normalization, and a P = 45% dropout. The method was integrated with the pre-activation batch normalization method for proper network training, computational stability, and convergence speed.

Table 2. Result of the label count

Label Name	Label Count
Facial Expressions	880
Makeup	880
Occlusion	880
Old Age Faces	880
Pose	880
Younger Age Faces	880

The proposed model yielded the label count result shown in Table 2 above. The model can figure out how many images are in each class. The label count is a table containing the labels for each class and the number of images [8]. The model is capable of detecting, categorizing, and calculating the number of images in each of the six classes.

Table 3. Shows the details of the proposed CNN training plot

Parameter	Value
Trained Accuracy	100.00%
Test Accuracy	99.87%
Status	Completed
Completion Time	28 min 08 sec
Number of Epoch	80
Number of Iteration	880
Number of Frequency	90

Fig. 6 is the training progress graph. The first graph at the top represents training accuracy (classification accuracy). The x-axis represents a scale of 0 to 80 epochs and 0 to 880 iterations, respectively, while the y-axis represents an accuracy value scale of 0% to 100%. The loss function (cross-entropy loss errors) is depicted in the bottom graph. This graph shows the iteration used to estimate the gradient. A light blue line and a dark blue line represent the classification accuracy. This accuracy was obtained by implementing a smooth approach to training accuracy.

Table 3 shows the data from the experiment's training plot. Fig 7 shows the training accuracy, validation accuracy, training loss, and validation loss.

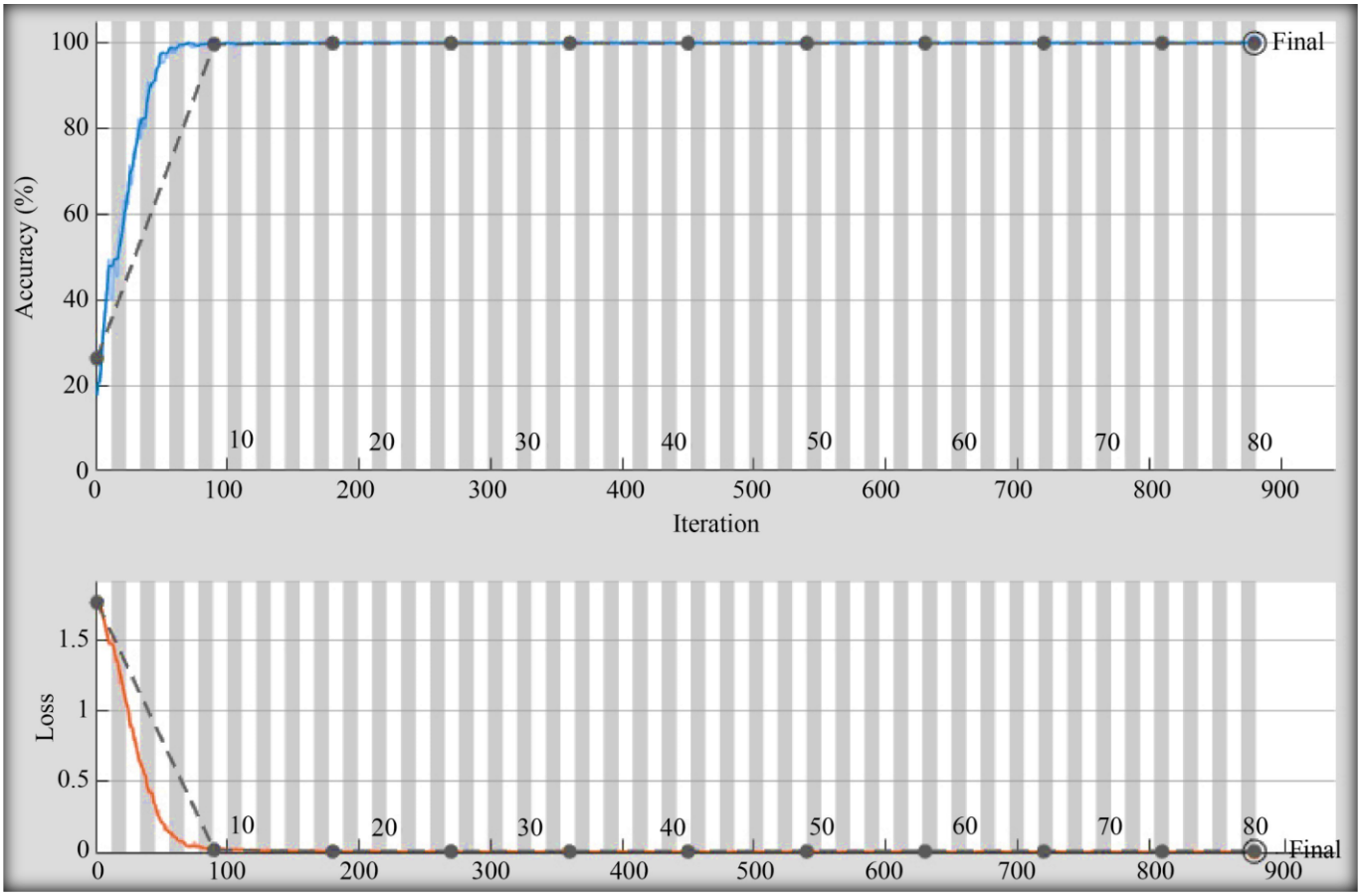

Fig. 6 Training progress plot graph of the proposed 22 layers CNN

The interrupted black dotted line defines the classification accuracy of the entire validation dataset. The number of iterations of each epoch is represented by the dotted black markings (points) on the black dashed line. The study has set the model to 90 iterations per epoch on the graph.

It is clear that the model began to converge around epoch 9, iteration 90, with training accuracy nearing 100%. Moreover, validation accuracy reached 99.62%. The network continued to converge successfully until epoch 28, iteration 300 when the training accuracy dropped to 99.75%, and the validation accuracy increased to 99.87%. With the power of batch normalization and dropout in the MBCRA, as well as the dropout in the fifth convolutional unit, the training accuracy regained its convergence rate with 100.00% accuracy at epoch 32 and iteration 350, and the validation accuracy rate was maintained at 99.87%. The training and validation accuracy of 100.00% and 99.87% remained unchanged in the last epoch, 80, and iteration 880.

All of this was accomplished in 28 minute and 8 seconds of processing time by MBCRA's regularization operation. At the bottom of the second graph is the loss function. The light orange line represents the training loss, while the dark dotted line represents the validation loss. The broken line represents the loss on each mini-batch and the loss on the validation dataset [9]. The number of images used for training and validation is 70% for training and 30% for validation, with 70% for training and 30% for validation (testing). The training and validation loss functions (the light orange and dark dotted lines) converged to the minimum error at the last epoch of 80 and iteration 880, respectively, with training loss values of 0.0020 and validation loss values of 0.0075 [10]. This shows that both the middle blocks convolutional regularization algorithm (MBCRA) and the pre-activation batch normalization algorithm (PABNA) used in the model architecture work well since the proposed batch normalization, dropout, and ReLU activation function hybrid model consistently gains very low training and validation loss error rates and achieves immediate convergence speed.

9.1.1. A Model Performance Evolution Matrix

Sometimes the deep learning models give out an accuracy of 90%, 91%, up to 99%, and so on, but this is not what it needs to depend on from the models' given accuracy. Because sometimes that does not reflect the actual truth of the result.

Table 4. Performance evaluation result for the proposed multilayers CNN+SVM based on 6 classes of variations

Deep Learning Models	*Classes of Variations*	Performance Evaluation			F_1-Score
		Accuracy	Precision	Sensitivity (Recall)	
Multilayer CNN+ SVM	**Facial Expression**	1.00	1.00	1.00	1.00
	Makeup	1.00	1.00	1.00	1.00
	Occolusion	1.00	1.00	1.00	1.00
	Old Age	0.9988	1.00	0.9923	0.9961
	Pose	1.00	1.00	1.00	1.00
	YongerAge	1.00	1.00	1.00	1.00
	Average	0.9998	1.00	0.9987	0.9994

To appraise the true performance of the proposed hybrid multilayer CNN+SVM architecture on each of the six classes of face variation, this study adopted an evaluation metric evaluation measures of accuracy, precision, recall, and f1-score to evaluate and prove the good performance of the proposed model on the multi-class classification of six human face image versions. The model accuracy is achieved by dividing the correctly classified samples by all the samples.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (13)$$

$$\text{Recall} = \frac{TP}{TP+FN} \quad (14)$$

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (15)$$

$$F_1\text{-Score} = \frac{2(\text{Precision})*(\text{Recall})}{\text{Precision}+\text{Recall}} \quad (16)$$

Where TP, TN, FP, and FN denote true positive, true negative, false positive, and false negative, respectively [11].

Table 4 displays the performance evaluation results of the proposed hybrid multilayer CNN+SVM based on each class of face image variations (uncontrolled conditions), including facial expiration, facial makeup, occlusion, old age, pose, and young age variations. The results were obtained using four different measurement standards, namely accuracy, precision, recall, and f1-score. From the table, it can be seen that the best value among all the measurements is precision. So the result shows higher precision and a higher recall. This implies that the proposed model avoids making excessive mistakes during classification, indicating that it makes reliable predictions and is robust. The evaluation table clearly shows that the suggested multilayer CNN+SVM has a robust feature extraction and classification capability, with an average accuracy of 0.9998, precision of 1.00, recall of 0.9987, and f1-score of 0.9994. The f1-score shows that both precision and recall are in balance, as the f1-score has a value of 0.9994, which is close to 1.

Generally, the proposed model has better performance in terms of all the measurement standards. Above all, the proposed model has good performance for face recognition applications under different uncontrolled conditions. This means the proposed model is resilient to all six facial image variation classes involving facial expressions, facial makeup, occlusion, age-related variation, and pose.

9.2. Experiment with 15 Layers of CNN

The second experiment was conducted using a 15-layer CNN model. This model was developed and used by [26]. In their research, they used the model to enhance the 2D face recognition system to learn discriminating representation. This model consists of a single input layer of size 64x64x3 and three convolutional units. Each convolutional unit has one convolutional layer, followed by a BN layer, a Relu function, and a max-pooling layer. The model has one fully connected layer, one soft-max layer, and one output layer. This research uses this model to compare its performance with the proposed multilayer CNN+SVM model.

Table 5 shows the training information for the 15-layer CNN. These are details about the training and validation phases. It can be seen from the table that the training completion time is 28 min 55 sec at a maximum number of epochs of 80 and iterations of 880.

Table 5. Training plot detail for the 15-layer CNN training from scratch

Parameters	Values
Trained Accuracy	99.25%
Testing Accuracy	98.69%
Status	Finished
Processing Time	28 min 55 sec
No. of epoch	80
No. of Iteration	880
Frequency	90 iteration

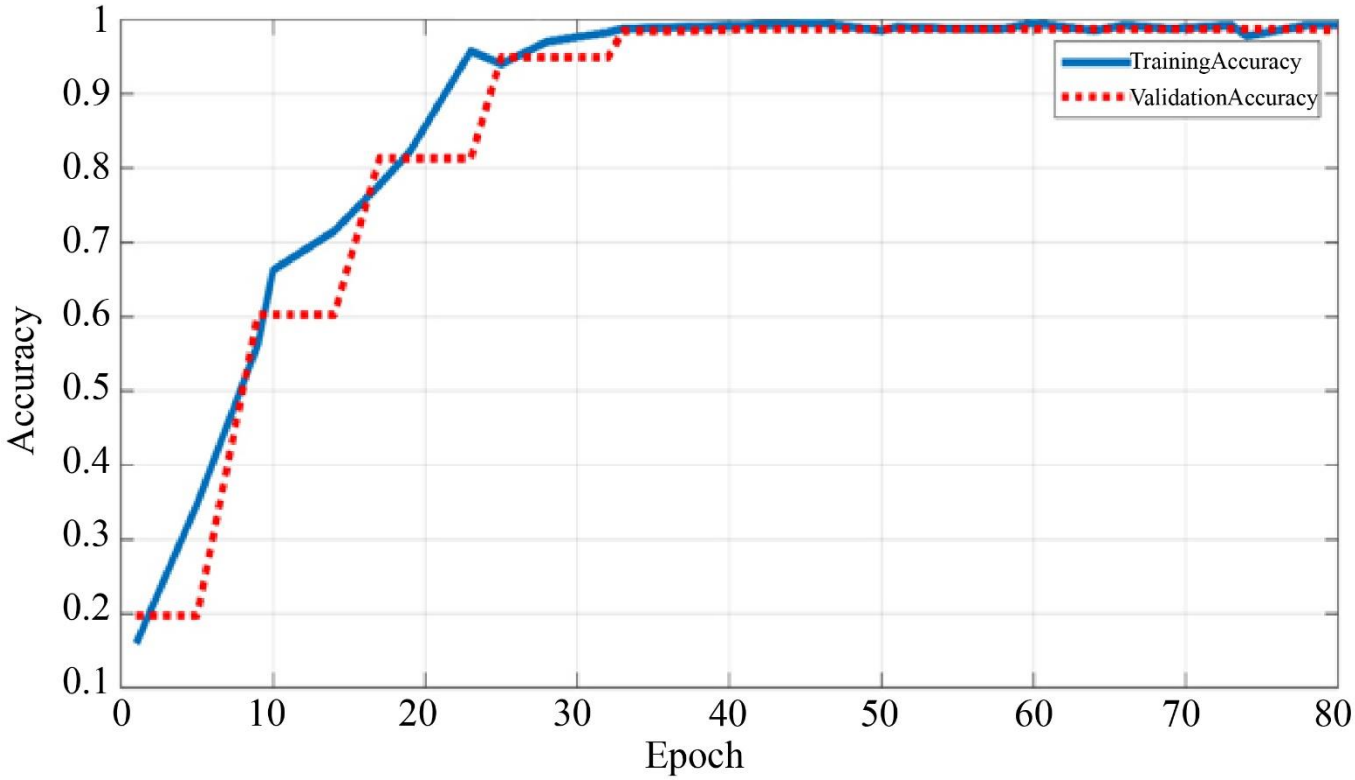

Fig. 7 Training and validation accuracies (Classification accuracies curve) for 15-layer CNN

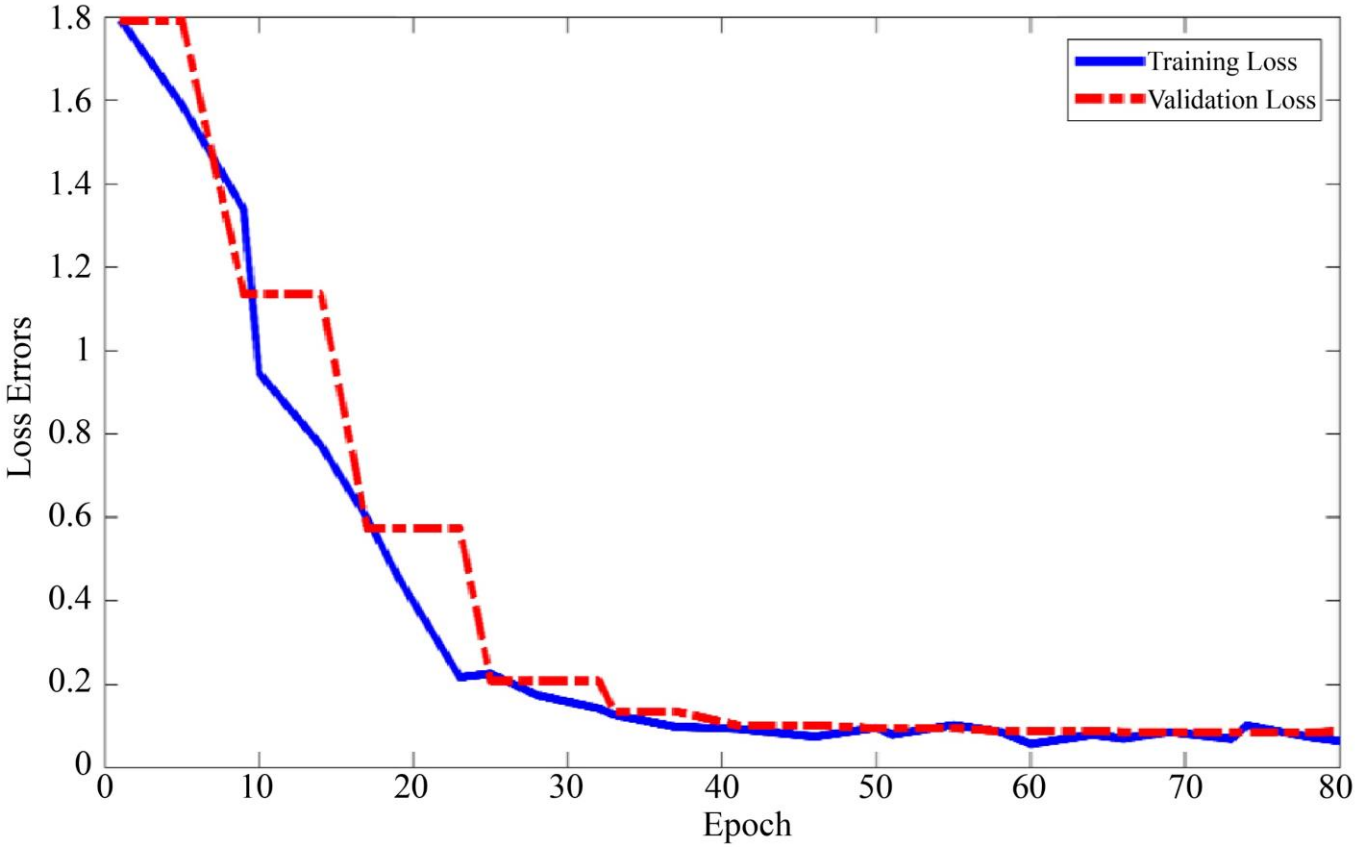

Fig. 8 Training and validation losses (Cross entropy curve) for 15 LYCNN

Table 6. Performance evaluation result for 15 layers based on 6 classes of variations

Deep Learning Models		Performance Evaluation			
	Classes of Variation	Accuracy	Precision	Sensitivity (Recall)	F_1-Score
15 layers CNN	Facial Expression	0.9976	0.9857	1.00	0.9931
	Makeup	0.9976	0.9857	1.00	0.9927
	Occolusion	0.9940	0.9786	0.9856	0.9821
	OldAge	0.9976	0.9928	0.9928	0.9958
	Pose	0.9976	0.9857	0.9718	0.9786
	YoungerAge	0.9940	0.9928	0.9720	0.9823
	Average	0.9964	0.9868	0.9870	0.9874

Figure 7 demonstrates that a notable observation may be made at the start of the model's training throughout the first few epochs (1 to 28) and iterations (1 to 300). Both the training and validation accuracies (solid blue and dotted red lines) begin with higher volatility due to overfitting. Both the training and validation accuracy plateaued for a short period at epoch 32 until the gradient updates escaped the unfavorable local minimum and began to converge at epoch 41 and iteration 450 with training and validation accuracies of 99.25% and 98.45%, respectively. At epoch 50, the training accuracy dropped to 98.50% until epoch 66, when it restored its typical accuracy value of 99.25%, while at the same epoch 66, the validation accuracy increased to 98.65%.

The model kept these accuracy levels up until the last epoch of 80 and iteration 880. As a result of the additional batch normalization before the rectifier linear unit (Relu) function in three of the convolutional units, the model achieved greater overall training and validation accuracy. The model has achieved higher validation accuracy in facial expression, facial makeup, and pose variation datasets with an average procession of about 0.9868 and a recall of 0.987. As illustrated in table IIX.

In Fig. 8, the loss error curve shows that the training (solid blue line) and validation (dotted red line) loss errors, which describe the degree of discrepancy between model prediction and real classes, generate a larger plateau of fluctuation of gradient transmitted through the network caused by gradient overfitting. It can be observed that the model's gradients are generally higher at the beginning of training and gradually decrease to minimize errors.

The performance of the 15-layer CNN architecture was verified. The accuracy, precision, recall, and f1-score performance evolution metrics were utilized to evaluate the model based on each face variation prediction class, including facial exhalation, facial makeup, occlusion, old age, pose variation, and young age. The table shows the best and lowest classes based on the accuracy, precision, and recall measurement standards and the f1-score for each variation class. It can be seen that the model has the best values for prediction in the class of OldAge, with an accuracy value of 0.9976, precision of 0.9928, recall of 0.9928, and an f1-score value of 0.9958. Here, both precision and recall are higher, and since the two are higher, it means that the model is not making mistakes in classifying the predicted class. However, facial expression, facial makeup, and pose variation have the same accuracy value of 0.9976 and the same precision value of 0.9857 with different recalls and f1-score values.

So, in this case, the f1-score values are used to determine the best-predicted class among the entire variation classes. This is because the f1-score measures the positive and negative predictions in an equal way. It is the harmonic mean between precision and recall. So the best-predicted classes are the classes of old age variation, with an f1-score of 0.9958, the class of facial expiration, with an f1-score of 0.9931; and facial makeup, with an f1-score value of 0.9927. The model has low predictive power when it comes to posing, occlusion, and YoungAge variations. Though their f1-score values have reached 0.9786, 0.9823, and 0.9821, respectively, the values are lower than those of the other three classes. This shows that the model is resistant to classes of OldAge, facial expressions and facial makeup.

9.3. Experiment with 9-Layers CNN&SVM

In this experiment, 9 layers of CNN and SVM were combined to recognize human faces. This model architecture was developed by [15]. The convolutional neural network is used as a feature extractor to get remarkable features, at the same time as the SVM is employed as a classifier. The model architecture consists of nine hidden layers: one input layer, three convolutional layers, three max-pooling layers, one fully connected layer, and one output layer. Each neuron in each feature map is attached to a 3x3 local receptive field in all three convolution layers. Table 7 contains information about the model training phases.

Table 7. Training plot details for the 9 layers CNN+SVM training from scratch

Parameters	Values
Trained Accuracy	99.50
Testing Accuracy	99.49
Status	Complete
Processing Time	30 minutes 49 seconds
No. of epoch	80
No. of Iteration	11
Frequency	90

Table 7 displays the details of the training progress plot graph in relation to the model training phase. The table shows that the training completion time is 30 minutes and 49 seconds, with a maximum number of iterations of 880.

Fig. 9 shows the model accuracy in training (a solid light green line) and testing (a dotted reddish line), showing the change in model training and testing performances over 80 training epochs and 880 training iterations. The above plot suggests that the performance for the training forms a smooth plateau at around epochs 1 to 10, while the testing performance is bad and worsens at epochs 19 and 23. While the performances for both training and testing begin to converge at roughly 99.25% and 98.97% at epoch 37 and iteration 400, respectively, at epoch 50 and iteration 550, the model appears to learn the problem swiftly and converges to the local minimum. This holds true until the last epoch (80) and iteration (880), with training and testing accuracies of around 99.50% and 99.49%, respectively.

In Figure 10, the training and validation loss functions are set to 1.7921 and 1.7988, respectively. At epoch 50, the training and validation loss values are 0.0257 and 0.0314, respectively. The training loss value is lowered to 0.0176 and 0.0306 at epoch 58. The final training and validation loss values climb to 0.0216 and 0.0297 at epoch 80. Because the training and validation loss values differ significantly, these results show that the 9-layer CNN model has no overfitting problem.

The performance evaluation results for the 9-layer CNN+SVM in Table 8 show that the prediction based on accuracy, precision, recall, and f1-score is significantly superior in three classes of variants. The three classifications are posture variation (with accuracy and f1-score values of 0.9987 and 0.9961, respectively), occlusion (accuracy and f1-score values of 0.9974 and 0.9923, respectively), and face makeup (with an accuracy value of 0.9974 and an f1-score value of 0.9922).

Table 8. Performance evaluation result for 9 layers CNN+SVM on the classes of variation

Deep Learning Models		Performance Evaluation			
	Classes of Variations	**Accuracy**	**Precision**	**Sensitivity (Recall)**	**F_1-Score**
9 layers CNN& SVM	Facial Expression	0.9948	0.9923	0.9772	0.9847
	Makeup	0.9974	0.9922	1.00	0.9922
	Occolusion	0.9974	1.00	0.9848	0.9923
	OldAge	0.9910	0.9769	0.9694	0.9731
	Pose	0.9987	0.9923	1.00	0.9961
	YongerAge	0.9948	0.9692	1.00	0.9844
	Average	0.9957	0.9872	0.9873	0.8218

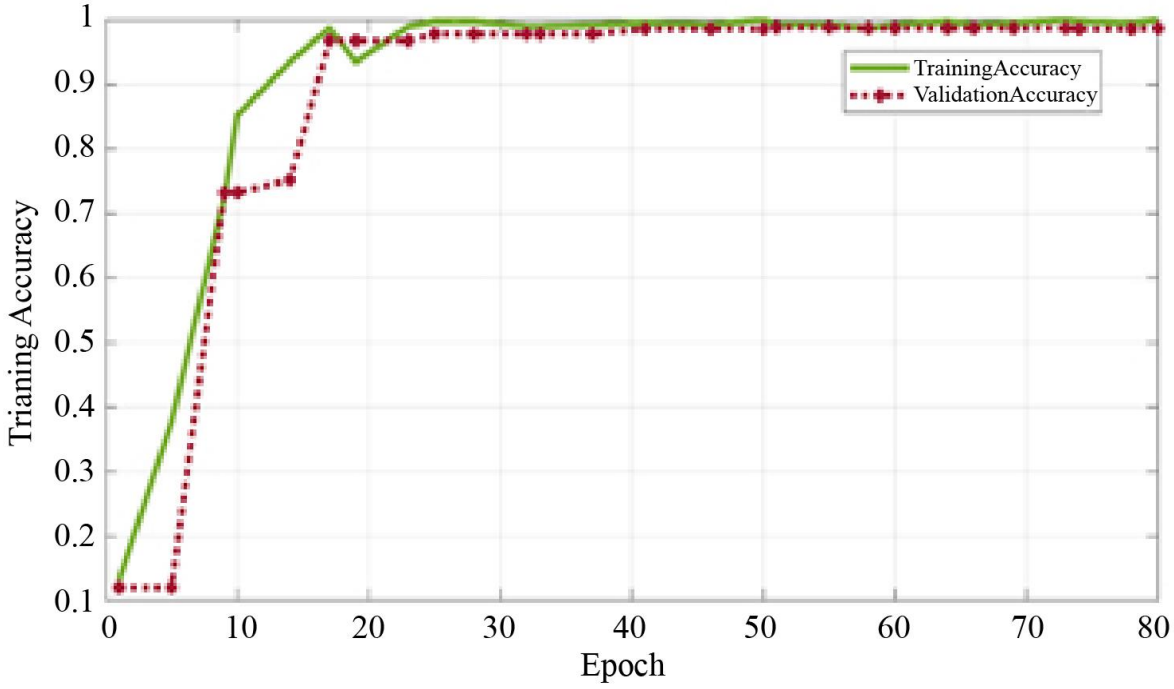

Fig. 9 Training and validation accuracies (Classification accuracies curve) for 9-layer CNN+SVM

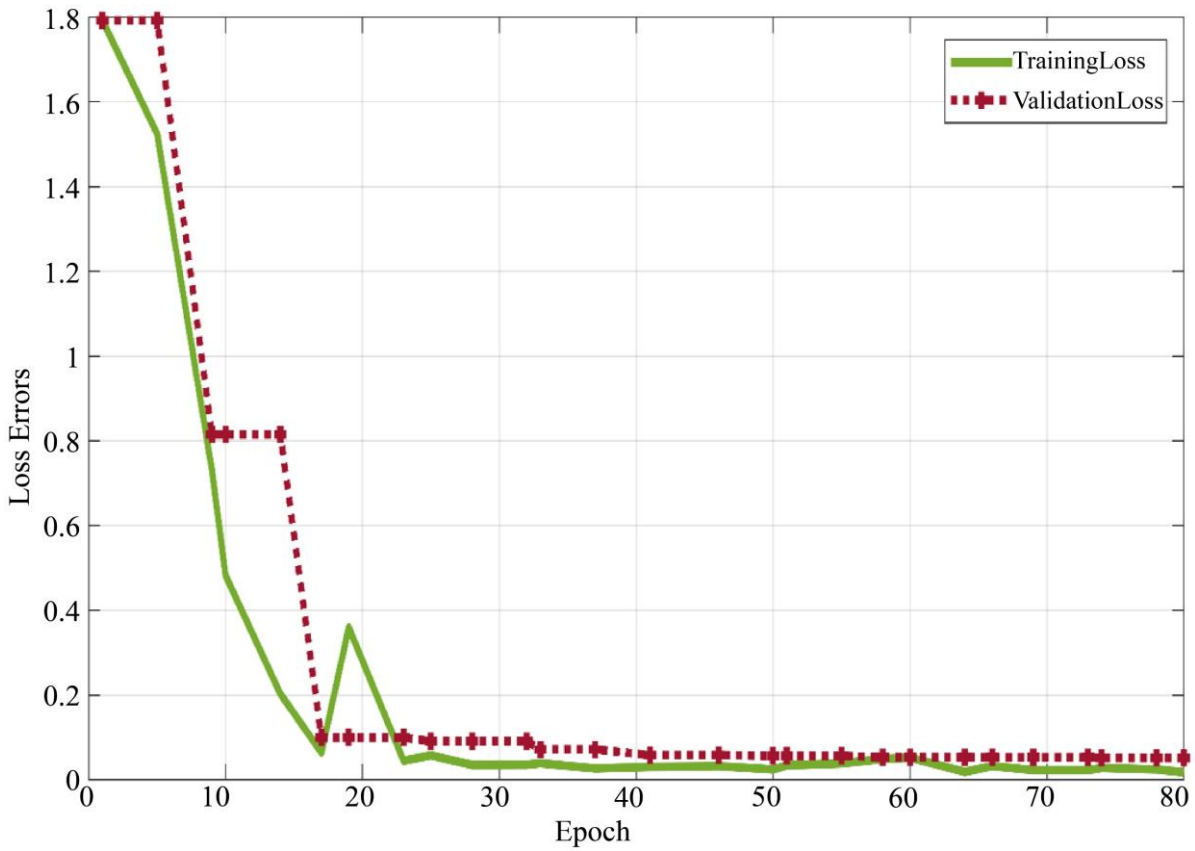

Fig. 10 Training and validation losses (Cross entropy curve) for 9 LYCNN

The model performs poorly on facial expiration, old age, and young age variations, all with lower prediction values when compared with the above results of the other three classes. Thus, generally, the model has better classification performance in the class of pose variation, with a higher accuracy value of 0.9987, prediction value of 0.9923, recall of 1.00, and an f1-score value of 0.991.

9.4. Experiment with 7 Layers CNN

The performance of the 7-layer CNN model for face recognition was evaluated by [51]. This model has one input layer, two convolutional layers, two max-pooling layers, and two fully connected layers as its main structure. Each neuron in the feature map is connected to a 3x3 local receptive field in each of the two convolution layers, and each neuron in the two max-pooling layers is linked to a 2x2 local receptive field in the preceding layer in each of the two max-pooling layers. As for its activation function, the model relied on a rectifier linear unit (ReLu). The model training techniques are detailed in Table 9, which displays the data for the training progress plot graph. The table shows that the training completion time is 21 minutes and 29 seconds at 80 epochs and 880 maximum iterations.

In the graph above, the model training and testing performances are set at 16.00% and 38.21%, respectively, at epoch 1. At epoch 41, the performances climbed dramatically to 100.00% and 99.49%, respectively. After epoch 60, the training accuracy drops to around 99.75%, while the validation accuracy remains at 99.49%.

The ultimate training and testing accuracies were 100.00% and 99.49%, respectively, at epoch 80. Throughout the training process, the testing dataset's accuracy is close to that of the training dataset. This indicates that there was no overfitting and that the regularization approaches (batch normalization layer, dropout layer, and data argumentation techniques) were effective. [15] address the issue of minor errors in encapsulating structure installation and production. Furthermore, the inaccuracies are difficult to quantify, resulting in lenses with non-parallel optical axes. The final experimental results show that this strategy is feasible.

Table 9. Training Plot Details for 7 Layers CNN&SVM, Training from Scratch

Parameter	Value
Trained Accuracies	100.00%
Validated Accuracies	98.97%
Train Status	Completed
Completion Time	29 min 21 seconds
Number of epoch	80
Mux. Number of Iteration	11
Validation Frequency	90

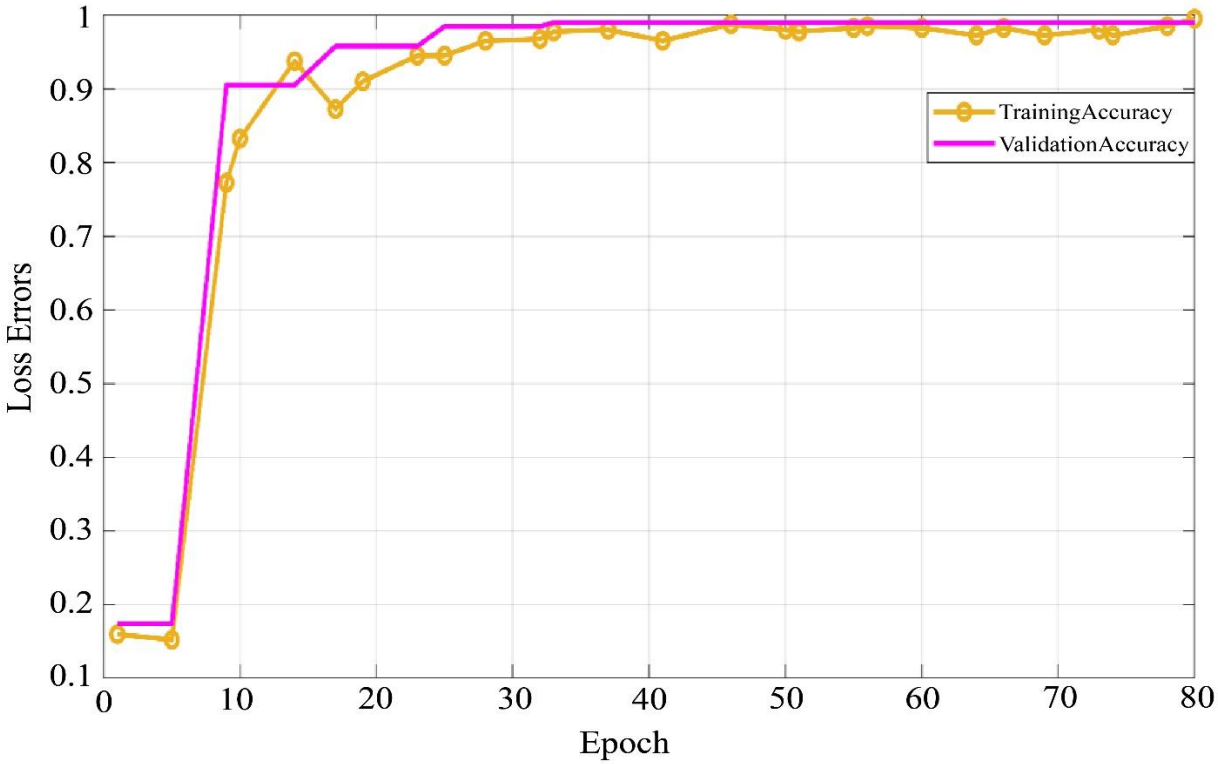

Fig. 11 Training and validation accuracies (Classification accuracies curve) for 7-layer CNN

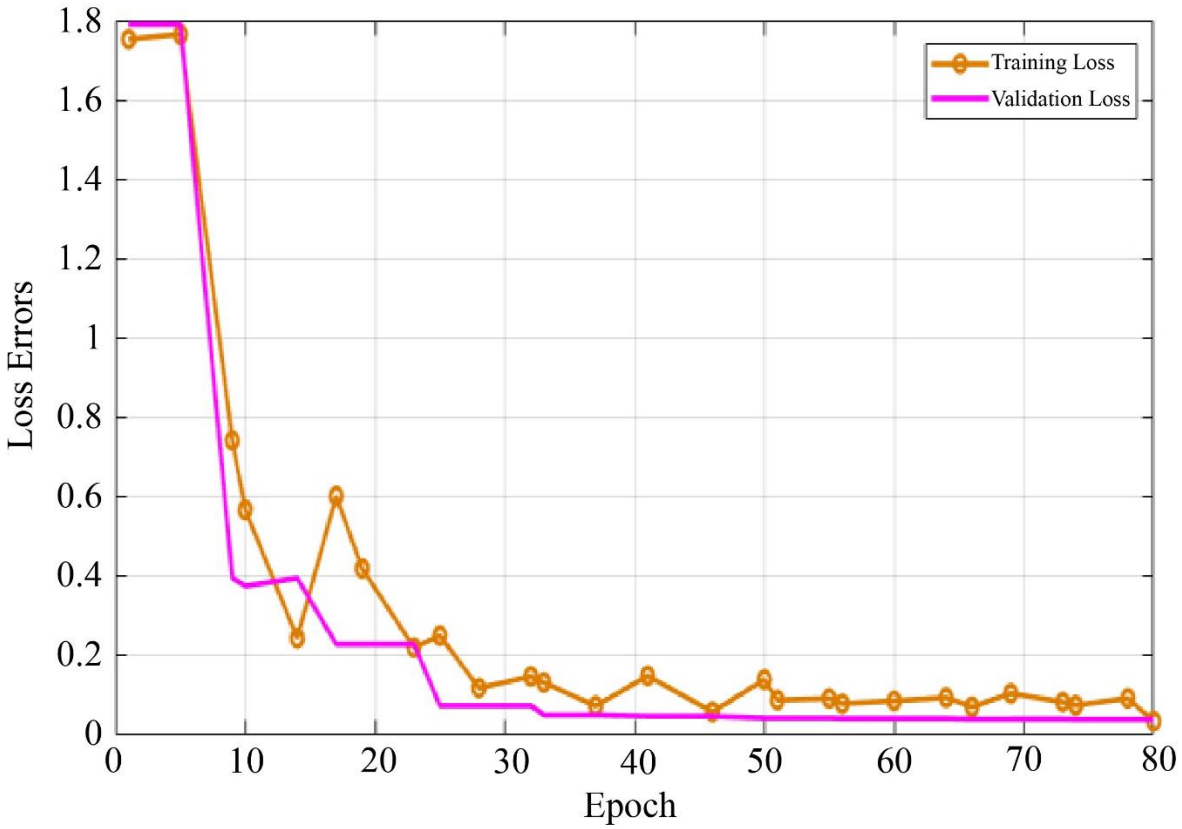

Fig. 12 Training and validation losses (Cross entropy curve) for 7 LYCNN

Table 10. Performance evaluation result for 7 Layers CNN based on 6 classes of face image variation

Deep Learning Models		Performance Evaluation			
	Classes of Variations	Accuracy	Precision	Sensitivity (Recall)	F_1-Score
7 layers CNN	Facial Expression	0.9936	0.9615	1.00	0.9804
	Makeup	0.9974	0.9846	1.00	0.9922
	Occolusion	0.9897	0.9385	1.00	0.9682
	OldAge	0.9679	1.00	0.9091	0.9524
	Pose	0.9885	0.9462	0.9840	0.9647
	YoungerAge	0.9832	0.9538	0.9920	0.9725
	Average	0.9867	0.9641	0.9809	0.9717

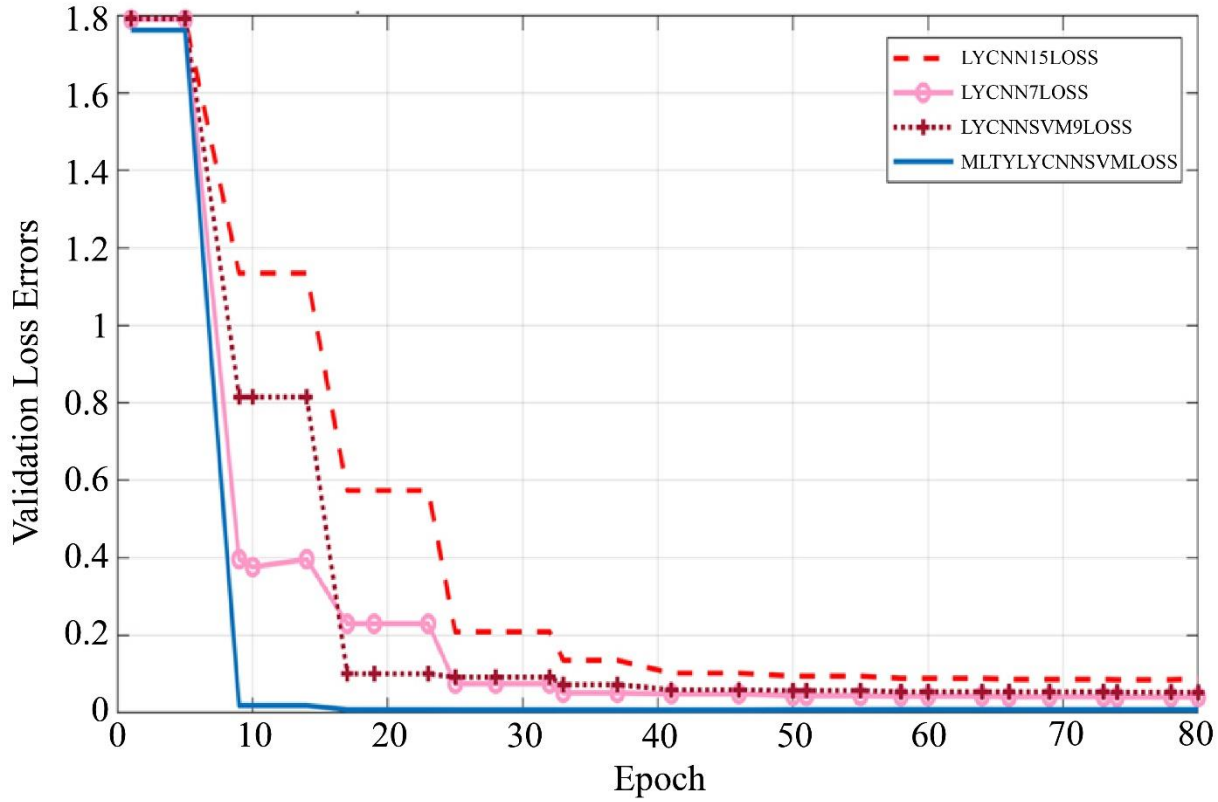

Fig. 13 Validation loss comparison between the proposed MLTYLY CNN and three deep learning model architectures

Figure 12 depicts a line plot of the training loss (solid pink line) and validation loss (dotted orange line), reflecting the degree of discrepancy between the model prediction and the true classes, which decreases as the epoch number increases [15]. At epoch 1 and iteration 1, the training and losses were 1.8011 and 1.7360, respectively.

At this point, both the training and testing datasets complicate model convergence. With the power of the MBCRA, the model eventually stabilizes and escapes from the massive flotation. The loss errors begin to drop to the local minimal error around epoch 33 and iteration 360, with training and validation loss errors of about 0.0340 and 0.0318, respectively. The training and testing losses are 0.0226 and 0.0271, respectively, when the number of epochs reaches the ultimate stage of 80 epochs and 880 iterations [16].

The performance measures were used to assess the model's ability to predict accurately in each of the six classes of face image variations. The exact findings of the models' good and poor forecasts are shown in Table 10.

According to the table, the model is more resilient in making strong predictions in the category of facial makeup, with an accuracy value of 0.9972, a precision value of 0.9846, a recall value of 1.00, and fi-score values of 0.9922.

The model's lowest prediction is in the old age category, with an accuracy value of 0.9679 and an f1-score of 0.9524. As a result, when compared to the other five classes, the model is more robust in producing appropriate classifications of facial makeup, with an accuracy of 99.72%.

Table 11. Comparative analysis of face recognition ability of different deep learning models for overall performances

Deep Learning Models	Performance Evaluation				
	Average Accuracy	Average Precision	Average Recall	Average F_1-Score	Processing Time
Proposed MLYCNN+SVM	0.9998	1.00	0.9987	0.9994	28: 08
15-LYCNN [17].	0.9964	0.9868	0.9870	0.9874	28:55 sec
9-LYCNN (Guo et al., 2017).	0.9957	0.9872	0.9873	0.8218	30:49 sec
7-LYCNN [19].	0.9867	0.9641	0.9809	0.9717	21:29 sec

9.5. Comparative Analysis of the Four Models Performance Evaluation Results

In this section, the study explicates the details of the performance evaluation results obtained from the four deep learning architectures. To test and compare the performance of the proposed Multilayer CNN+SVM architecture with that of the other three deep learning models. The study still used the four performance evaluation metrics: accuracy, precision, recall, and F1-score. The loss error performance graph curve in Figure 12 was also used to demonstrate the good performance of the proposed hybrid model.

$$\text{Loss} = \frac{1}{m} * \sum_{i=1}^{m} yi\ logf(xi) \qquad (14)$$

Where m is the value of training images xi and yi are the input and expected output, respectively, $f(xi)$ denoted the real output.

Fig. 13 shows the comparison line curve for validation loss errors between the proposed hybrid multilayer CNN (solid blue line) and three other deep learning architectures that involve 15-layer CNN (dash red line), 9-layer CNN+SVM (dotted marron line), and 7-layer CNN (solid bobble pink line). The proposed hybrid model architecture has the four model architectures' lowest and most stable loss error. More specifically, the proposed model starts to be more stable at epoch 17 and lasts up to epoch 80.

This means the model has an earlier convergence at epoch 17 with an error value of 0.0075, while it continues to be stable to the last epoch with an error value of 0.0075. These results show that the proposed model has the representational power to overfit the gradient disappearance problem since the validation loss values have continued to have the same loss values around zero values right from the beginning [2].

This indicates that the regularization effect of both the MBCRA and PABNM is strong enough to prevent the proposed model from overfitting. While the other three models still exhibit dramatic fluctuations up to the 80th epoch, more specifically, the 15-layer CNN has a higher error value of 0.0872. This model has an overlapping loss error; it conjectures that the model has nowhere near the representational power to classify any of the face images in the six classes of the dataset correctly compared to the other three models due to its overlapping error. It indicated that the model is unstable and has a higher overfitting problem. Generally, it can be stated that the proposed hybrid model performs well in classifying all six classes of face variations.

The performance evaluation results for each of the four deep learning models are shown in Table 11. From the table, it can be seen that the best value among all the measurements is the average precision of the proposed hybrid multilayer CNN+SVM, with a value of 1.00. This signified that the model avoids making many mistakes during classification, indicating that the proposed model is making correct predictions and is robust. In this section, the evaluation table clearly indicates that the proposed multilayer CNN+SVM has a robust feature extraction and classification capability compared to the other three models. It can be seen that the proposed model has a high average accuracy of 0.9998, average precision of 1.00, an average recall of 0.9987, and an average f1-score value of 0.9994. The f1-score indicates that precision and recall are balanced since it has a value of 0.9994, which is close to 1.

The lowest performance result, on the other hand, goes to the 7-layer CNN, which has an average accuracy of 0.9867, an average precision of 0.9641, a recall of 0.9809, and an f1-score of 0.9717. Overall, the proposed model performs better across all measurement standards [54]. Furthermore, the proposed model performs well in face recognition applications under various uncontrolled conditions, most notably in the categories of facial expressions, cosmetics, occlusion, old age, posture, and younger age. From the comparative analysis table above, it can be noticed that the proposed MLYCNN+SVM has a training completion time of 28 min 08 seconds, which is 0.47 sends faster than the 15-layer CNN and 2 min 41 seconds faster than the 9-layer CNN. However, the proposed model is 6 min 76 seconds slower than the 7-layer CNN, with a training completion time of 21 min 29 seconds. When considering the number of their internally connected layers, the completion time of the proposed model is almost the same as that of the 7 layers of CNN [55].

10. Conclusion

This research presents an improved and efficient framework for human face recognition applications based on a hybrid deep learning technique involving multilayer convolutional neural networks (CNN) and support vector machines (SVM). The research analyzes the effects of

regularization strategies by incorporating a new MBCRA into the proposed model's design. The study shows that multi-layer CNN+SVM is more effective when using a batch normalization layer between a convolutional layer and a dropout, with a 45% probability of dropout in the middle convolutional unit. This is important when using large training and testing datasets.

The performance evaluation result obtained from each of the four models shows that the proposed model is more robust for face image classification under all six unconstrained conditions when compared with the 15-layer CNN, which has a problem in classifying face images under occlusion with an accuracy of 0.9950 and young age variation with an accuracy of 0.9940. When compared with 7-layer CNN and 9-layer CNN, these two deep learning models have the problem of classification under facial expressions, older age, and young age variations. Deep learning methods with multiple convolutional connections can extract more complicated facial characteristics.

Acknowledgement

We would like to appreciate and applaud the efforts of the University Sultan Zainal Abidin, particularly the Faculty of Informatics and Computing, as well as the UNISA Project Management Centre (CRIEM), for the academic and financial support of this research. We also thank the management of Al-Qalam University Katsina, Katsina State, Nigeria. For providing fellowship support to the Corresponding Author.

References

[1] A. Vinay et al., "Face Recognition Using Gabor Wavelet Features with PCA and KPCA - A Comparative Study," *Procedia Computer Science*, vol. 57, pp. 650–659, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[2] Syafeeza Ahmad Radzi et al., "A MATLAB-Based Convolutional Neural Network Approach for Face Recognition System," *Journal of Bioinformatics, Proteomics and Imaging Analysis*, vol. 2, no. 1, pp. 1–5, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[3] Mejda Chihaoui et al., "Face Recognition Using HMM-LBP," *Hybrid Intelligent Systems*, vol. 420, pp. 249-258, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[4] Mohammed Bennamoun, Yulan Guo, and Ferdous Sohel, "Feature Selection for 2D and 3D Face Recognition," *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 1–28, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] Lei Chen et al., "Face Recognition with Statistical Local Binary Patterns," *International Conference on Machine Learning and Cybernetics*, pp. 2433–2439, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[6] Omkar M. Parkhi, Andrea Vedaldi, and Andrew Zisserman, "Deep Face Recognition," *Proceedings of the British Machine Vision Conference (BMVC)*, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[7] Tsung-Yi Lin et al., "Microsoft COCO: Common Objects in Context," *European Conference on Computer Vision*, vol. 8693, pp. 740–755, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[8] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 815–823, 2015. [Google Scholar] [Publisher Link]

[9] Sanusi Darma Abu, and Fatma Susilawati Mohamad, "Approaches of Deep Learning in Persuading the Contemporary Society for the Adoption of New Trend of AI Systems: A Review," *International Journal Of Scientific & Technology Research*, vol. 9, no. 12, pp. 163–177, 2020. [Google Scholar] [Publisher Link]

[10] Zahraddeen Sufyanu et al., "Feature Extraction Methods for Face Recognition," *International Review of Applied Engineering Research (IRAER)*, vol. 5, no. 3, pp. 5658-5668, 2016. [Google Scholar] [Publisher Link]

[11] Tejaswi Satepuri, and P. Chandrasekar Reddy, "A Survey on Facial Expression Recognition Techniques," *International Journal of Computer Sciences and Engineering*, vol. 7, no. 5, pp. 980–984, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[12] Umara Zafar et al., "Face Recognition with Bayesian Convolutional Networks for Robust Surveillance Systems," *EURASIP Journal on Image and Video Processing*, vol. 2019, no. 10, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[13] Anandhavalli Muniasamy, and Areej Alasiry, "Deep Learning: The Impact on Future eLearning," *International Journal of Emerging Technologies in Learning*, vol. 15, no. 1, pp. 188–199, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[14] A. M. Turing, "Computing Machinery and Intelligence," *The Mind Association*, vol. LIX, no. 236, pp. 433–460, 1950. [CrossRef] [Google Scholar] [Publisher Link]

[15] Shanshan Guo, Shiyu Chen, and Yanjie Li, "Face Recognition Based on Convolutional Neural Network and Support Vector Machine," *IEEE International Conference on Information and Automation (ICIA)*, pp. 1787-1792, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[16] Firoz Mahmud et al., "PCA and Back-Propagation Neural Network-Based Face Recognition System," *18th International Conference on Computer and Information Technology (ICCIT)*, pp. 582–587, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[17] Mohannad Abuzneid, and Ausif Mahmood, "Improving Human Face Recognition Using Deep Learning Based Image Registration and Multi-Classifier Approaches," *IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)*, pp. 1-2, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[18] Peibo Duan et al., "Applying DCOP to User Association Problem in Heterogeneous Networks with Markov Chain Based Algorithm," *arXiv, Multiagent Systems*, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[19] Mohannad A. Abuzneid, and Ausif Mahmood, "Enhanced Human Face Recognition Using LBPH Descriptor, Multi-KNN, and Back-Propagation Neural Network," *IEEE Access*, vol. 6, pp. 20641–20651, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Li Guo et al., "Face Image Classification Using Appearance and Texture Features," *International Conference on Computer Application and System Modeling*, pp. 476–480, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[21] Matthew C. Fysh, and Markus Bindemann, "Human-Computer Interaction in Face Matching," *Cognitive Science*, vol. 42, no. 5, pp. 1714-1732, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[22] LeCun et al., "A B7CEDGF HIB7PRQTSUDGQICWVYX HIB edCdSISIXvg5r CdQTw XvefCdS," *Proceeding IEEE*, 1998. [Google Scholar]

[23] Teofilo F. Gonzalez, *Handbook of Approximation Algorithms and Metaheuristics*, 1st Edition, Chapman and Hall/CRC, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[24] Mohammed Kamel Benkaddour, and Abdennacer Bounoua, "Feature Extraction and Classification Using Deep Convolutional Neural Networks, PCA and SVC for Face Recognition," *Traitement du Signal*, vol. 34, no. 1–2, pp. 77–91, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[25] T. Kujani, and V. Dhilip Kumar, "Emotion Understanding from Facial Expressions using Stacked Generative Adversarial Network (GAN) and Deep Convolution Neural Network (DCNN)," *International Journal of Engineering Trends and Technology*, vol. 70, no. 10, pp. 98-110, 2022. [CrossRef] [Publisher Link]

[26] Samar S. Mohamed et al., "Deep Learning Face Detection and Recognition," *International Journal of Electronics and Telecommunications*, pp. 1–7, 2019. [Google Scholar] [Publisher Link]

[27] Hayder Najm, Hayder Ansaf, and Oday A. Hassen, "An Effective Implementation of Face Recognition Using Deep Convolutional Network," *Journal of Southwest Jiaotong University*, vol. 54, no. 5, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[28] Urvashi Bakshi, and Rohit Singhal, "A Survey on Face Detection Methods and Feature Extraction Techniques of Face Recognition," *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, vol. 3, no. 3, pp. 233–237, 2014. [Google Scholar] [Publisher Link]

[29] Mohammed Abdallh Otair, and A. Salameh Walid, "Efficient Training of Backpropagation Neural Networks," *Neural Network World*, vol. 16, no. 4, pp. 291–311, 2015. [Google Scholar] [Publisher Link]

[30] Hesham M. Eraqi, Mohamed N. Moustafa, and Jens Honer, "End-to-End Deep Learning for Steering Autonomous Vehicles Considering Temporal Dependencies," *Machine Learning, arXiv*, pp. 1–8, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[31] A. R. Syafeeza et al., "Convolutional Neural Network for Face Recognition with Pose and Illumination Variation," *International Journal of Engineering and Technology*, vol. 6, no. 1, pp. 44–57, 2014. [Google Scholar] [Publisher Link]

[32] Yaniv Taigman et al., "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701-1708, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[33] Vivienne Sze et al., "Efficient Processing of Deep Neural Networks: A Tutorial and Survey," *Proceedings of the IEEE*, vol. 105, no. 12, pp. 2295–2329, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[34] Wei Wang et al., "Development of Convolutional Neural Network and its Application in Image Classification: A Survey," *Optical Engineering*, vol. 58, no. 4, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[35] Moacir Antonelli Ponti et al., "Everything You Wanted to Know About Deep Learning for Computer Vision but Were Afraid to Ask," *SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*, pp. 17–41, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[36] J. Wolfe et al., "Application of Softmax Regression and its Validation for Spectral-Based Land Cover Mapping," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, pp. 455–459, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[37] S.H. Shabbeer Basha et al., "Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image Classification," *Neurocomputing*, vol. 378, pp. 112–119, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[38] Gary B. Huang et al., "Labeled Faces in the Wild : A Database for Studying Face Recognition in Unconstrained Environments," *Artificial Intelligence*, pp. 1–11, 2008. [Google Scholar] [Publisher Link]

[39] Imokhai T. Tenebe et al., "Bacterial Contamination Levels and Brand Perception of Sachet Water: A Case Study in Some Nigerian Urban Neighborhoods," *MDPI Water*, vol. 15, no. 9, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[40] Norah Alnaim, Maysam Abbod, and Rafiq Swash, "Recognition of Holoscopic 3D Video Hand Gesture Using Convolutional Neural Networks," *Technologies*, vol. 8, no. 2, p. 19, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[41] Nicole Christoff et al., "Morphological Crater Classification via Convolutional Neural Network with Application on MOLA data," *IEEE Advances in Neural Networks and Applications*, pp. 1-5, 2018. [Google Scholar] [Publisher Link]

[42] Bernd Fritzke "A Self-Organizing Network that Can Follow Non-Stationary Distributions," *Artificial Neural Networks — ICANN'97*, vol. 1327, pp. 613–618, 1997. [CrossRef] [Google Scholar] [Publisher Link]

[43] Changzhi Bai et al., "Classification of Gas Dispersion States via Deep Learning Based on Images Obtained from a Bubble Sampler," *Chemical Engineering Journal Advances*, vol. 5, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[44] Onome Christopher Edo et al., "Why do Healthcare Workers Adopt Digital Health Technologies- A Cross-Sectional Study Intergrating the TAM and UTAUT Model in a Developing Economy," *International Journal of Information Management Data Insights*, vol. 3, no. 2, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[45] Joyassree Sen et al., "Face Recognition Using Deep Convolutional Network and One-shot Learning," *SSRG International Journal of Computer Science and Engineering*, vol. 7, no. 4, pp. 23-29, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[46] Cheng Xing, Jie-Sheng Wang, and Bo-wen Zheng, "Hybrid Face Recognition Method Based on Gabor Wavelet Transform and VGG Convolutional Neural Network with Improved Pooling Strategy," *IAENG International Journal of Computer Science*, vol. 48, no. 2, pp. 1–14, 2021. [Google Scholar] [Publisher Link]

[47] Ting-ting Yang, Su-yin Zhou, and Ai-jun Xu, "Rapid Image Detection of Tree Trunks Using a Convolutional Neural Network and Transfer Learning," *IAENG International Journal of Computer Science*, vol. 48, no. 2, pp. 1–8, 2021. [Google Scholar] [Publisher Link]

[48] Abu Sanusi Darma, and Susilawati Fatima Mohmad, "The Regularization Effect of Pre-Activation Batch Normalization on Convolutional Neural Network Performance for Face Recognition System Paper," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 11, pp. 300–310, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[49] Zahraddeen Sufyanu, Fatma Susilawati Mohamad, and Ahmad Salihu Ben-Musa, "A Proposed Integrated Human Recognition for Security Reassurance," *American Journal of Applied Sciences*, vol. 12, no. 2, pp. 155–165, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[50] Mokhairi, Makhtar et al., "Comparison of Image Classification Techniques Using Caltech 101 Dataset," *Journal of Theoretical and Applied Information Technology*, vol. 71, no. 1, pp. 79–86, 2015. [Google Scholar] [Publisher Link]

[51] Patrik Kamencay et al., "A New Method for Face Recognition Using Convolutional Neural Network," *Advance in Electrical and Electronic Engineering*, vol. 15, no. 4, pp. 663-672, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[52] M. Rajeshwari, and K. Rathika, "Palm Print Recognition Using Texture and Shape Features," *International Journal of Computer Science and Engineering*, vol. 9, no. 2, pp. 1–5, 2022. [CrossRef] [Publisher Link]

[53] D. J. Samatha Naidu, and R. Lokesh, "Missing Child Identification System using Deep Learning with VGG-FACE Recognition Technique," *SSRG International Journal of Computer Science and Engineering*, vol. 9, no. 9, pp. 1-11, 2022. [CrossRef] [Publisher Link]

[54] D. M. Leppinen, and S. B. Dalziel, "Bubble Size Distribution in Dissolved Air Flotation Tanks," *Journal of Water Supply: Research and Technology-Aqua*, vol. 53, no. 8, pp. 531–543, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[55] Raja Durratun Safiyah et al., "Performance Evaluation for Vision-Based Vehicle Classification Using Convolutional Neural Network," *International Journal of Engineering & Technology*, vol. 7, pp. 86–90, 2018. [Google Scholar] [Publisher Link]

[56] R. Lemlich, "Foam Fractionation and Allied Techniques," *Industrial & Engineering Chemistry Research*, vol. 60, no. 10, pp. 16–29, 2015. [Google Scholar]