Article

A Hybrid Method of Enhancing Accuracy of Facial Recognition System Using Gabor Filter and Stacked Sparse Autoencoders Deep Neural Network

Abdullah Ghanim Jaber [1], Ravie Chandren Muniyandi [1,*], Opeyemi Lateef Usman [1] and Harprith Kaur Rajinder Singh [2]

[1] Research Centre for Cyber Security, Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, UKM Bangi 43600, Selangor, Malaysia
[2] Faculty of Data Science and Information Technology, INTI International University, Nilai 71800, Malaysia
* Correspondence: ravie@ukm.edu.my

Abstract: Face recognition has grown in popularity due to the ease with which most recognition systems can find and recognize human faces in images and videos. However, the accuracy of the face recognition system is critical in ascertaining the success of a person's identification. A lack of sufficiently large training datasets is one of the significant challenges that limit the accuracy of face recognition systems. Meanwhile, machine learning (ML) algorithms, particularly those used for image-based face recognition, require large training data samples to achieve a high degree of face recognition accuracy. Based on the above challenge, this research proposes a method for improving face recognition precision and accuracy by employing a hybrid approach of the Gabor filter and a stacked sparse autoencoders (SSAE) deep neural network. The face image datasets from Olivetti Research Laboratory (OLR) and the Extended Yale-B databases were used to evaluate the proposed hybrid model's performance. All face image datasets used in our experiments are grayscale image type with a resolution of 92×112 for the OLR database and a resolution 192×168 for the Extended Yale-B database. Our experimental results showed that the proposed method improved face recognition accuracy by approximately 100% for the two databases used at a significantly reduced feature extraction time compared to the current state-of-art face recognition methods for all test cases. The SSAE approach can explore large and complex datasets with minimal computation time. In addition, the algorithm minimizes the false acceptance rate and improves recognition accuracy. This implies that the proposed method is promising and has the potential to enhance the performance of face recognition systems.

Keywords: Gabor filter; face recognition; deep neural network; stacked sparse autoencoders; hybrid method

1. Introduction

Face recognition systems played an important role in the human verification process to eliminate unauthorized user access in various applications. The users are verified with the help of an ID verification process in which the user's facial features are stored in the database to complete the user authentication. Facial identification enhances overall security in applications such as e-banking, e-commerce, forensics, airport security, etc. [1,2]. Face recognition aims to give a computer system the ability to quickly and precisely recognize human faces in images or videos [3,4]. Numerous algorithms and methods, including recently proposed deep learning models, have been proposed to improve face recognition performance [5–7]. However, the face recognition system is far from perfect in terms of accuracy.

Meanwhile, the environment in which face recognition is used influences its accuracy. Various factors influence face recognition accuracy, particularly unconstrained face

recognition, because face images exhibit multiple variations. These factors include pose variation, scale variation, partial occlusion, and complex illumination, which may impede recognition accuracies [1,8].

Most researchers use different techniques and algorithms to locate facial features. In addition, large-scale identification methods are incorporated to explore the facial features to maximize facial recognition accuracy. Learning ability, variety, and generalization are all advantages of deep neural network-based recognition algorithms [9–11]. When real-time operation is necessary and in unconstrained situations, efficient algorithms still have significant constraints due to the high accuracy and processing efficiency requirement [2]. Therefore, face recognition remains a considerable challenge in real-time applications, and it is a hot research topic in computer vision, deep learning, real-time systems, and other fields. The study uses the Gabor filter and deep learning model to maximize the facial recognition rate. The Gabor filter analyzes the captured images which are effectively utilized to perform the image textures. This method maximizes interpretability and discrimination tracker performance. In addition, it is able to locate the features related region with minimal computation difficulties. Therefore, this research uses the Gabor filter to extract related textural features from the input image. The extracted features are further investigated with the help of the Stacked Sparse Autoencoder (SSAE) deep neural model to identify the authenticated user. The main intention of this work is to create a robust and flexible system to recognize face images while trying to access the data in the database. The proposed hybrid Gabor filter and deep learning models effectively investigate the facial images, and verification is performed. During the analysis, the system uses the OLR dataset and the Extended Yale-B Face image dataset to evaluate the proposed system's efficiency. The information in the databases was captured with the help of different emotions and directions that help to recognize the user's facial expression with a minimum false acceptance rate. These databases were captured using excellent mobile and multimedia technology environments. This leads to an effective image dataset with a huge volume of images. These images are more helpful in evaluating the produced SSAE system's performance in order to achieve the stated research objectives. Hence, this study uses the hybridized deep neural model (SSAE) to maximize face recognition accuracy.

The rest of the manuscript is arranged as follows: Section 2 describes the materials and methods adopted in this work to achieve the stated objectives. Section 3 presents the experimental results and discussion, followed by an evaluation of the proposed hybrid method of the Gabor filter and SSAE system's efficiency. Section 4 describes some important research findings in detail. Section 5 concludes the paper and discusses the future direction of these works.

2. Materials and Methods

2.1. Materials

The collection of standard facial datasets for benchmarking purposes was a critical component of the consistent advancements in facial expression and expression recognition. In the 1990s, different techniques and methods were introduced by various researchers to maximize facial recognition accuracy. Numerous facial recognition databases currently contain face images that differ in terms of expressions, conditions, size, occlusions, poses, number of images, and lighting. The two most popular of these databases were used in this study.

1. The first database is the OLR, which contains a collection of face images photographed between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge, UK. This database can be accessed via https://cam-orl.co.uk/facedatabase.html (accessed on 10 June 2021). Accordingly, each of the 40 distinct human subjects has ten different facial photographs. The photos were taken at different times and with various facial details (no glasses/glasses) and facial appearance (non-smiling/smiling, closed eyes/open eyes). All photographs were taken against a dark homogeneous backdrop, with subjects standing frontally, upright, tolerating any rotation, and tilting

up to about 20 degrees. There are some variations in the scale range, of up to 10%. Figure 1 depicts some face image samples from the OLR database. These images are grayscale image-type and have a resolution of 92 × 112 pixels. In order to reduce computation time, we resized the selected face images in the OLR database by half of their original sizes in this work.

(**a**)   (**b**)
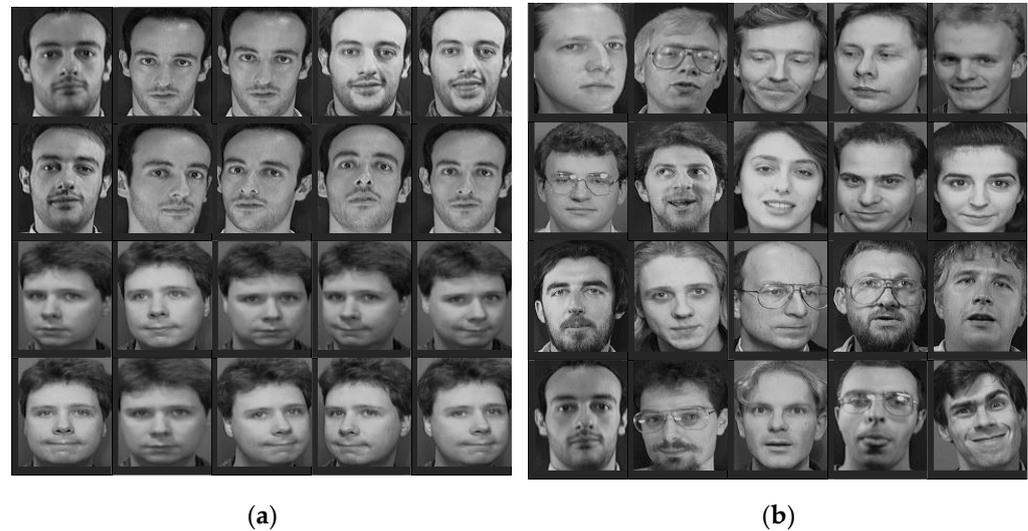
Figure 1. Sample face images (**a**) different poses of two people (**b**) different poses of various people.

2. However, the second database used in this study is the Extended Yale-B database. This database contains 2432 frontal face images, each with a dimension of 192 × 168 pixels for all the 38 human subjects. This database can be accessed at http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html (accessed on 10 June 2021). Furthermore, each subject has 64 photographs with varying levels of illumination. The photographs were taken under various lighting intensities and facial expressions. The intensity of lighting on these faces varies greatly across subjects, to the point where only a small portion of the face is visible in some cases. We close-cropped these face datasets with each photograph cropped to include only a look without hair or background. In addition, we resized the face images to half of their original sizes in order to reduce the computation time of the proposed model. Figure 2 shows face image samples from the Extended Yale-B database.

(**a**)   (**b**)
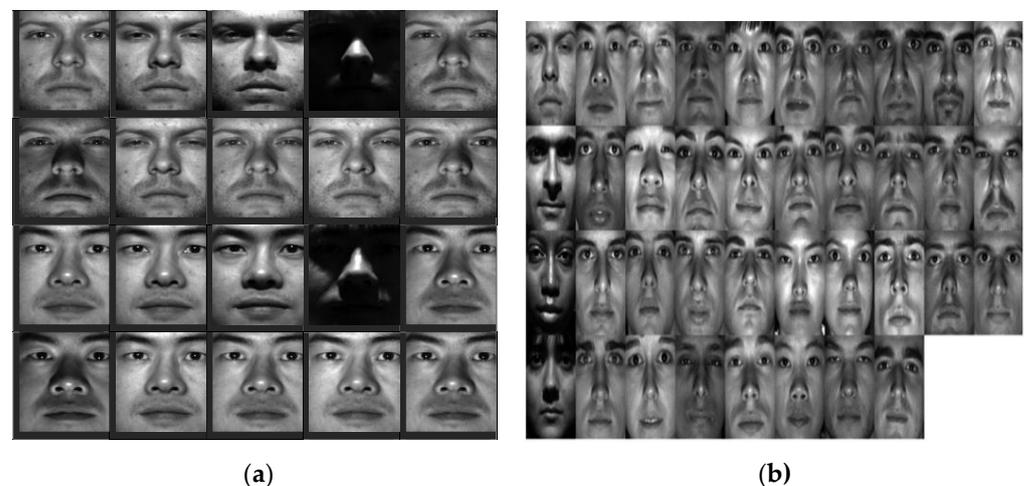
Figure 2. Sample images from the Extended Yale-B database (**a**) through two poses (**b**) from all poses.

Furthermore, a more detailed description of these two datasets can be found in Table 1, which contains the properties of these two-dimensional (2D) face datasets. The image differences are signified by (*i*) illumination, (*t*) delay time, and (*p*) pose.

Table 1. Characteristics of databases used.

Database	RGB Color/Grayscale	Images Size	No. of Persons	No. of Images per Person	Variation	Description
OLR	Grayscale	92 × 112 pixel	40	10	*i, t*	- Dark background images - Restricted number of participants - Different lightening conditions, poses, emotions, directions
Extended Yale-B database	Grayscale	168 × 192 pixel	38	64	*p, i*	- Variation in 9 different poses - Illumination in 64 conditions

2.2. Methods

The introduced hybrid model has several stages of image noise removal in which images are resized into half of the original image size, Gabor filter-based feature extraction, and SSAE deep neural network-based face prediction. The proposed system working process is illustrated in Figure 3.
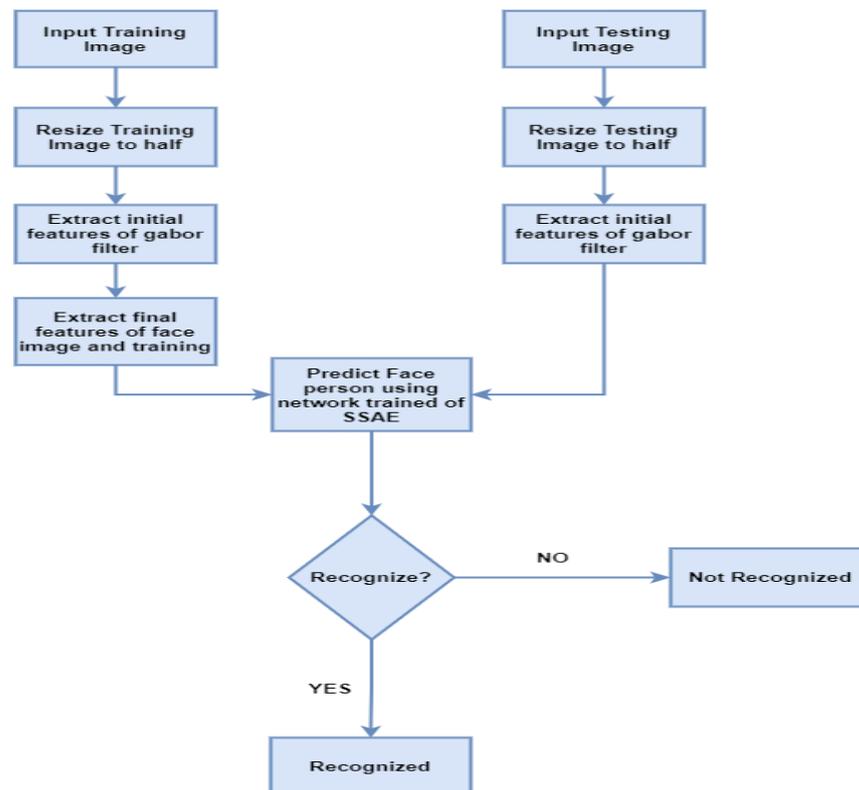
Figure 3. The hybrid deep neural model-based face recognition framework.

The proposed hybrid method of the face recognition system, as shown in Figure 3, combines two algorithms to achieve optimal results. These algorithms use the Gabor Filter and the Stacked Sparse Autoencoders (SSAE) CNN model for face recognition. The first step in reducing execution time is to resize the input images. Initially, the features are derived from the face images using the Gabor filter. The derived features were investigated with the help of the SSAE deep neural network model depicted in Figure 4. This study aims to improve facial recognition accuracy with minimal computation time.
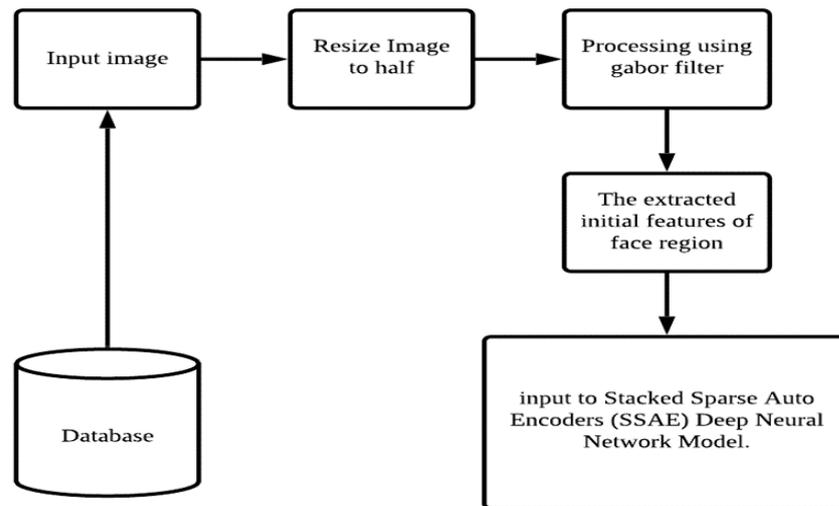
Figure 4. Structure of hybrid deep neural model.

2.3. Gabor Filters-Based Feature Extraction Method

Gabor filters (also known as Gabor wavelets) have properties similar to the human visual system, particularly for frequency and orientation representations. They are suitable for texture representation and discrimination. Gabor filters extract features directly from grayscale images using statistical information about character structures. However, in order to improve performance on low-quality images, the Gabor filter outputs are subjected to an adaptive sigmoid function [2,12–14]. A 2D Gabor filter is a complicated modulated sinusoidal function of a Gaussian kernel with a spatial response and frequency defined by Equations (1) and (2) (See Figure 5):

$$h(x,y;\lambda,\phi,\sigma,\sigma) = \frac{1}{2\pi\sigma_x\sigma_y}exp\left\{-\frac{1}{2}\left[\frac{R_1^2}{\sigma_x^2}+\frac{R_2^2}{\sigma_y^2}\right]\right\}\times exp\left[i\cdot\frac{2\pi R_1}{\lambda}\right] \tag{1}$$

where

$$R_1 = x\cos\phi + y\sin\phi$$

$$R_2 = -x\sin\phi + y\cos\phi$$

$$H(u,v;\lambda,\phi,\sigma_x,\sigma_y) = exp\left\{-2\pi^2\left(\sigma_x^2\left(F_1-\frac{1}{\lambda}\right)^2+\sigma_y^2(F_2)^2\right)\right\}\times C \tag{2}$$

where

$$F_1 = u\cos\phi + v\sin\phi$$
$$F_2 = -u\sin\phi + v\cos\phi \quad, C = \text{constant}$$

where Δx and Δy are denoted as the spatial localization of the Gabor filter that is computed with the help of spatial width, which is depicted in Equation (3).

$$(\Delta x)^2 = \frac{\int_{-\infty}^{+\infty} hh^*(R_1)^2 d(R_1)}{\int_{-\infty}^{+\infty} hh^* d(R_1)}$$

$$(\Delta y)^2 = \frac{\int_{-\infty}^{+\infty} hh^*(R_2)^2 d(R_2)}{\int_{-\infty}^{+\infty} hh^* d(R_2)} \tag{3}$$
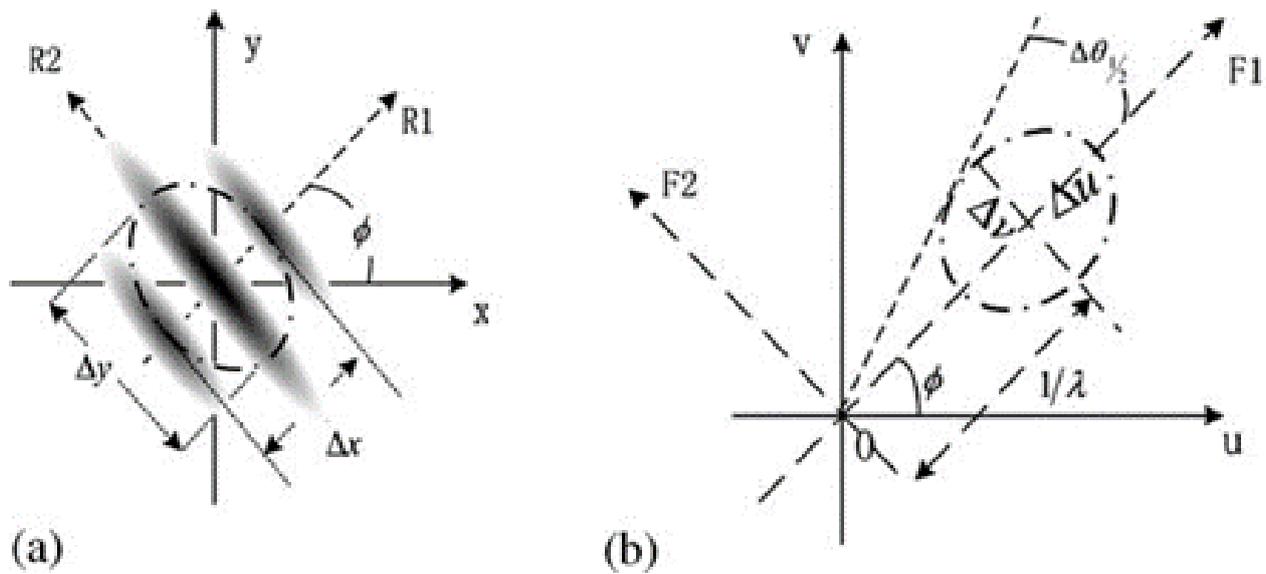
Figure 5. The proposed hybrid model steps. (**a**) spatial field viewpoint of a Gabor filter; (**b**) spatial frequency field.

$$\Delta x = \sigma_x/\sqrt{2}, \Delta y = \sigma_y/\sqrt{2} \tag{4}$$

Distances between Gabor filters adjacent to an image are mentioned as spatial sampling intervals and are defined by Dx and Dy, respectively. In order to avoid unintentional image data loss, the following relationships between effective spatial sampling interval and widths, as shown in Figures 5 and 6, must satisfy the condition shown in Equation (5):

$$\mathrm{D}x \leq \Delta x, \mathrm{D}y \leq \Delta y \tag{5}$$

Spatial sampling intervals are critical parameters to consider when designing a Gabor filter. However, in previous studies [15,16], it was not considered, resulting in poor performance and significant image detail loss. Gabor filter spatial-frequency localization can also be expressed using the efficient bandwidth measures Δv and Δu. In order to accomplish this, Equation (3) is transformed into Equation (6):

$$\Delta u = 1/\left(2\sqrt{2}\pi\sigma_x\right), \text{ and } \Delta v = 1/\left(2\sqrt{2}\pi\sigma_y\right) \tag{6}$$

Depending on spatial-frequency bandwidth, another concept known as the orientation bandwidth can be obtained [15], as indicated in Figure 7b.

$$\begin{aligned} \Delta\theta &\approx 2\arcsin((\Delta v_V/2)/(1/\lambda)) \\ &= 2\arcsin\left(\lambda/\left(4\sqrt{2}\pi\sigma_y\right)\right) \end{aligned} \tag{7}$$

In this study, we express spatial-frequency localization in 2D space in two different ways: line orientation selectivity and line-width selectivity, as depicted in Figure 7. During the analysis, $h(x, y : \lambda, \phi, \sigma_x, \sigma_y)$ is highly sensitive compared to the orientation $(\phi + \pi/2)$ with $\lambda/2$.
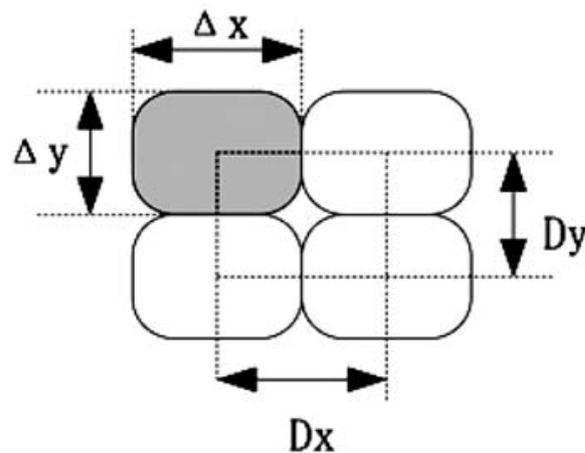
Figure 6. Representation of Gabor filtering width and Spatial Sampling.
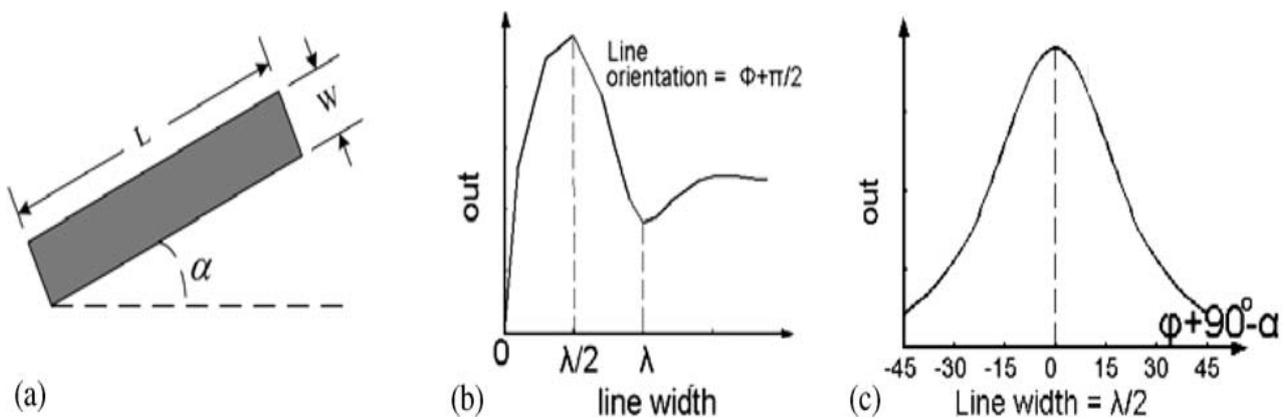
Figure 7. Length and width described in the above figure: (**a**) Gabor filter outputs changed concerning the width and orientations (**b**,**c**), implying Gabor filter selectivity of line width and orientation [17].

A feature extraction method based on Gabor filters is used to extract and locate initial features from the face region [17]. Gabor filters' main advantage is their resistance to translation, rotation, and scale. They also resist photometric disturbances such as lighting variations and image noise [2,12,18–20]. The Gabor filters' properties are extracted directly from grayscale photographs. As shown in Algorithm 1, a 2D Gabor filter is a Gaussian kernel controlled by a complex sinusoidal plane wave in the spatial domain.

Algorithm 1: An algorithm for the Gabor filter for feature extraction

---

Input: images after resizing
Output: feature regions and features in the image (length, width, orientations, frequency, bandwidth)
Initialization: f-sinusoid frequency, spatial aspect ratio γ, gaussian envelope σ and offset phase ϕ, Gabor function normal orientations θ
1: Read half-resized images with input values (f, π, γ, σ, and ϕ)
2: Estimate Gaussian function using:
$$G(x,y) = \frac{f^2}{\pi\gamma\eta} \exp\left(-\frac{x^2 + \gamma^2 y^2}{2\sigma^2}\right) \exp(j2\pi f x' + \phi)$$
3: Compute the Gaussian function values by using Equations (1) and (2):
4: Compute the image features according to the orientation and width values by using Equations (6) and (7).

The experiments are conducted on OLR database (56 × 46 image pixel) and the Extended Yale-B database (96 × 84 image pixel). During the analysis, 40 Gabor filters were applied in five different scales and eight different orientations. The description of these images are illustrated in Figure 8. The dimension of the feature vector for the OLR database using 40 Gabor filters is 56 × 46 × 40 = 103,040, while the size of the feature vector for the Extended Yale-B database is 96 × 84 × 40 = 322,560, because adjacent pixels in an image are frequently highly correlated.
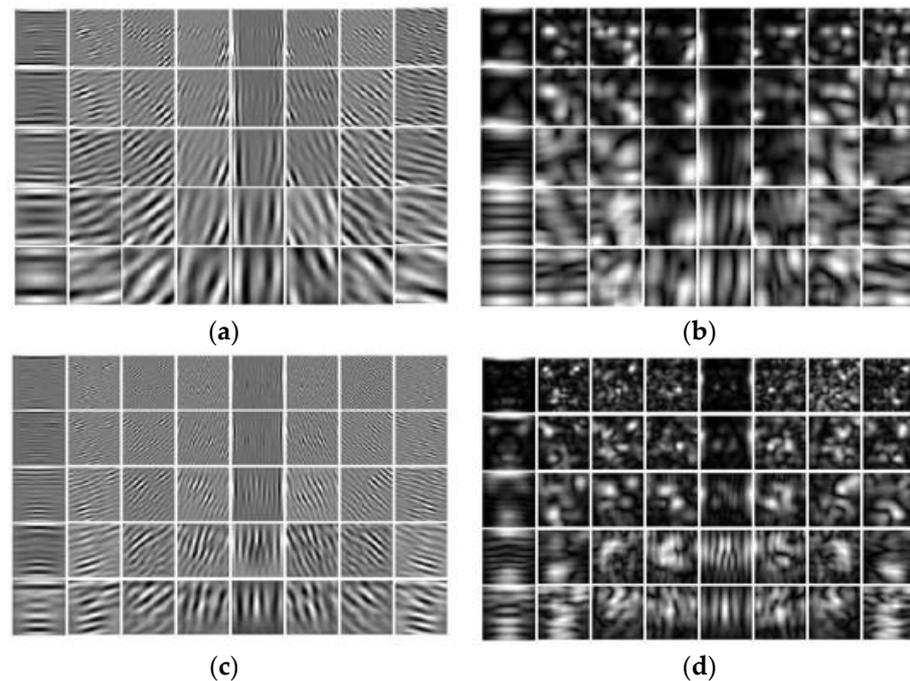
Figure 8. Gabor wavelet image representations: (**a**) OLR database images Gabor representation (**b**) OLR images magnitude representation (**c**) Extended Yale-B database representation and (**d**) magnitude value of Extended Yale-B database.

Furthermore, Gabor filter feature images can reduce information redundancy [12,19]. Downsampling feature images by a factor of sixteen yields a vector of 1680 in size for the OLR database and 5280 for the Extended Yale-B database. These vectors were also normalized to have a unit zero mean and variance. The derived Gabor filter facial features were then fed into the deep neural network model of stacked sparse autoencoders (SSAE).

2.4. Deep Neural Network and Autoencoders Model

Deep neural networks are feed-forward neural network derivatives with more than two hidden layers of highly connected neurons, and their training is referred to as deep learning models [10,11,21,22]. The multilayer feed-forward network, also known as the deep neural network, employs a lower unit number, with deep architecture to approximate complex functions with comparable accuracy. As a result, training parameters are reduced, allowing for training with relatively small datasets. The Autoencoder is one of these popular architectures [23,24].

The Autoencoder is one of the deep learning models used to learn the data features from the raw data. It has two units, an encoder and a decoder, which are used to compute the output value for the input parameters. The encoder has compresses that process the input value, and the decoder performs the opposite of the encoder's function. This algorithm's main intention is to maximize the data analysis rate, feature exemplification of the input, and effectively compute the dataset correlations. The Autoencoder utilizes the multiple-layer network working process to train the features and predicts the output value. In this study, the back-propagation learning algorithm [10,21,25] was used to prepare the features to reduce the deviation between the output values. In addition, the learning process was enhanced with the help of an encoder and a decoder, which help to update the network parameters such as weight w and bias b. The representation of the network is illustrated in Figure 9 [26].
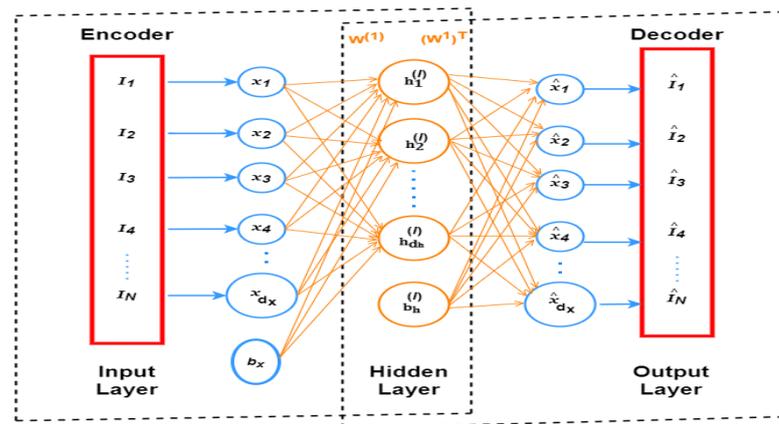
Figure 9. The simple sparse autoencoder architecture.

2.4.1. The Basic Sparse Autoencoder (SAE) Network

Suppose that $X = (x(1), x(2), \ldots, x(N))^T$ is the set of unlabeled initial face image features for training, where $x(k) \in R^{d_x}$, N and d_x is denoted as the number of pixels in the images, and the number of facial features is denoted as N. Then, the l-layer high-level learning features are computed using Equation (8) with kth features. During the computation d_h, hidden number units and current layer l are utilized.

$$h^l(k) = (h_1^l(k), h_2^l(k), \ldots, h_{d_h}^l(k))^T \tag{8}$$

Here, hidden neurons and units are defined using the superscript and subscripts. The 1st hidden layer of the ith unit is denoted as $h_i^{(1)}$ in Figure 8. Here, hidden layer l processes the x and $h^{(l)}$ number of features to identify the input image-related output value. In addition, the sparse autoencoder neural model is shown in Figure 9. The encoder has x inputs in the input layer and h hidden layer in the encoder that computes the outputs. Then, the decoder processes the input in h hidden layer to find the output. During this process, optimal parameters are utilized to reduce the deviations between the outcomes. The variations are reduced to minimize the output reconstruction. Therefore, the sparse autoencoder (SAE) is computed using Equation (9) [27].

$$\mathfrak{L}_{SAE}(\theta) = \left[\frac{1}{N}\sum_{k=1}^{N}\left(L\left(x(k), d_{\hat{\theta}}\left(e_{\widetilde{\theta}}(x(k))\right)\right)\right)\right] + \left[\alpha\sum_{j=1}^{n} KL(\rho||\hat{\rho}_j)\right] + \left[\beta||W||_2^2\right] \tag{9}$$

In Equation (9), the sum of the mean square error (SMSE) of the idiom that defines the contradiction among incoming $x(k)$ and rebuilding $\hat{x}(k)$ is the overhead of the entire set of data. Furthermore, $e_{\widetilde{\theta}}(\cdot)$ maps incoming $x \in R^{d_x}$ to the hidden illustration $h \in R^{h_x}$, which is computed via $h = e_{\widetilde{\theta}}(x) = s(Wx + b_h)$, where $b_h \in R^{d_h}$ in which bias b_h and W is a weights of the $d_h \times d_x$ matrix. The encoder is represented as $\widetilde{\theta} = (W, b_h)$ while decoder $d_{\hat{\theta}}(\cdot)$ plots outcoming hidden illustrations h back into the reconstruction space $\hat{x}$. $\hat{x} = d_{\hat{\theta}} = s(W^T h + b_x)$, where $b_x \in R^{d_x}$ is defined as bias and W^T is a $d_x \times d_h$ denoted as a weight matrix. $s(\cdot)$ is signified as an activation function; here, logistic sigmoid as $s(z) = \frac{1}{1+e^{-z}}$, was utilized as an activation function for neuron z. Therefore, the decoder is parameterized by $\theta = (W^T, b_x)$. The transposition of the matrix of weights W results in the matrix of weights W^T of the inverse designation. The Autoencoder successfully minimizes the weight matrix to half its size. The pre-activation of the output layers of the autoencoder, $\theta = (W, b_h, b_x)$, may be written as $y = W^T s(Wx + b_h) + b_x$ using three parameters. Therefore, the rebuilding of the decoder, $\hat{x}$, can be determined using $\hat{x} = s(y)$. The Autoencoder training aims to minimize the reconstruction error mentioned in the first phrase while optimizing the parameters $\theta = (W, b_h, b_x)$. The difference between the

incoming x and the reconstruction $\hat{x}$ made by the decoder $d_{\hat{\theta}}(\cdot)$ is determined by the cost function $L(\cdot, \cdot)$.

The second idiom uses the index j to represent the network's hidden unit total and the number n to represent the number of units in the hidden layer. Parameter $KL(\rho || \hat{\rho}_j)$ is the Kullback–Leibler (KL) divergence among $\hat{\rho}_j$, which defines the mean activation of hidden unit j (i.e., averaged activation over the training group) and desired activation ρ_j, described by Equation (10) as follows:

$$\rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} \tag{10}$$

The third idiom is a weight decay idiom, which employs Equation (11) to reduce the magnitude of the weight and helps to avoid overfitting:

$$||W||_2^2 = tr\left(W^T W\right) = \sum_{l=1}^{n_l} \sum_{i}^{s_{l-1}} \sum_{j}^{s_l} \left(w_{i,j}^{(l)}\right)^2 \tag{11}$$

where n_l is the number of layers and s_l is the number of neurons in layer l. In addition, $w_{i,j}^{(l)}$ demonstrates the connection among the ith neuron in $l-1$ as well as the jth neuron in l. In this study, the SAE are $n_l = 2$, and $s_1 = 1680$ for the OLR database and $s_1 = 5280$ for the Extended Yale-B database, $s_2 = 1200$.

2.4.2. Stacked Sparse Autoencoder (SSAE) Network

The SSAE consists of multiple simple SAE layers with their outputs linked to the following layer's inputs: a deep neural network. In this research, two fundamental SAEs are combined to produce two layers of SSAE. The design of the suggested SSAE deep neural network is shown in Figure 10.
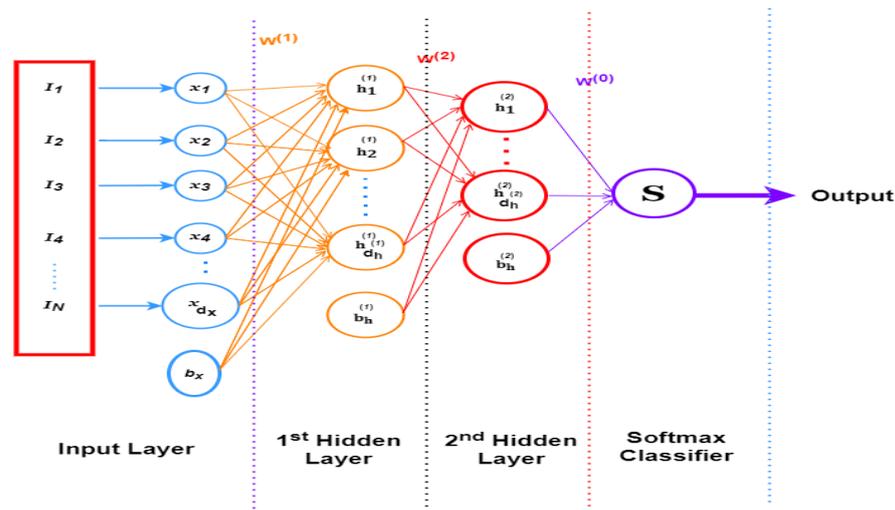
Figure 10. A proposed stacked sparse autoencoder (SSAE) architecture for face recognition with soft-max classifier [9,26].

The SSAE produces a function $f : R^{d_x} \to R^{d_{h^{(2)}}}$ that transforms the input pixels for the first face feature into a new feature exemplification, specified as: $h^{(2)} = f(x) \in R^{d_{h^{(2)}}}$. The input layer's vector of a column of pixel features describes the raw pixel of the initial facial picture feature. There are input units with $d_x = 1680$ for the Extended Yale-B database and $d_x = 5280$ for the OLR database in the input layer. The first and second hidden layers' hidden units are, respectively, $d_{h^{(1)}} = 1200$ and $d_{h^{(2)}} = 800$.

2.5. Training of the Proposed SSAE Deep Neural Network

We used the greedy layer-wise method for SSAE pre-training in order to train the proposed SSAE deep neural network for face recognition. This was achieved by introducing each layer individually. After pre-training, the trained SSAE was used to test the dataset set aside for extracting features for face classification. In order to learn basic features, an SAE first takes the inputs of a raw face x and a set of weights $W^{(1)}$. The network trained to generate the specific activations of feature $h^{(1)}(x)$ for each facial image feature x then receives its output. The sparse autoencoders use these fundamental features as "raw input face" to learn $h^{(2)}(x)$. The next activation function for feature $h^{(2)}(x)$ for each of the fundamental features $h^{(1)}(x)$, the features are then sent into the second SAE (which corresponds to the vital features of the initial face image features of the input x). Once a soft-max classifier has been trained to associate secondary features with number labels, secondary features are input.

The final step is to integrate all three layers to create SSAE, which has two hidden layers and a soft-max classifier that can accurately categorize the face traits from both the OLR and the Extended Yale-B database. Algorithm 2 presents the condensed training algorithm for the proposed SSAE deep neural network with a soft-max classifier.

Algorithm 2: An algorithm for training stacked sparse autoencoder (SSAE) model with soft-max classifier

---

Input: Extracted features by Algorithm 1
Output: Authenticated image or not
Initialization: bias b_h, weight W, input x.
1: Training of facial image features
// Training initial face image features using number of the pixels in each initial face feature
　2: Compute hidden layer output: $h = e_{\bar{\theta}}(x) = s(Wx + b_h)$
// where $b_h \in R^{d_h}$ is a vector of a bias, and W is a $d_h \times d_x$ matrix of weight
3: Calculate the next hidden layer output that will be used to predict the output value using Equation (8) as follows:

$$h^l(k) = \left(h_1^l(k), h_2^l(k), \ldots, \quad h_{d_h}^l(k) \right)^T$$

// where h is the input hidden layer
// enter feature of the initial face and its exemplification at hidden layer l
　4: Estimate the new feature-related output value f using: $f : R^{d_x} \rightarrow R^{d_{h^{(2)}}}$
// which convert pixels of input raw of initial face image feature to
　$h^{(2)} = f(x) \in R^{d_{h^{(2)}}}$ // new feature exemplification specified
　5: Estimate the soft-max optimization to predict the final output value using Equations (9)–(11).
// all three layers are merged jointly to shape SSAE with two hidden layers
// and an ultimate layer of soft-max classifier capable of classifying the face
//attributes of both the OLR and the Expanded Yale-B Face databases
　6: Input unrecognized:
　　　// returning to Algorithm 1 if more face photos are required for training.

---

3. Experimental Results and Discussion

All experiments were conducted using MATLAB (R2021b) software installed on a GPU-based system with a 2.70 GHz processor, 8.00GB RAM, and a 4 Core(s) Intel (R) processor due to the high-speed requirements. NVIDIA GeForce GTX680 is the GPU processor version used in the experiment system. This processor accelerates the development of deep neural network models. According to [11,28,29], GPU processors outperform CPU-based counterparts in terms of processing speed and memory usage. Face image features were extracted using the proposed 2D Gabor filters. The proposed SSAE deep neural network was trained on two hidden layers, using the extracted face image features from the OLR and the Extended Yale-B databases.

The proposed SSAE neural network model was trained on 2356 samples of initial face image features from the Extended Yale-B database and 320 representatives from the OLR databases. The initial input feature of face images in the Extended Yale-B database is 5280 pixels, while the initial input feature in OLR database is 1680 pixels. The training hyperparameters for the proposed SSAE deep neural network model are shown in Table 2.

Table 2. Training hyperparameters for proposed SSAE deep neural network.

Hyperparameters	Proposed SSAE Model for OLR	Proposed SSAE Model for Extended Yale-B Database
Training samples	320	2356
HL1 Size	1200	1200
HL2 Size	800	800
1st Autoencoder:		
Function for activation	Log-Sigmoid	Log-Sigmoid
Parameters of sparsity	0.15	0.15
Weight sparsity	4	4
Decay value of weight	0.004	0.004
Iterations (max)	400	400
2nd Autoencoder:		
Function for activation	Log-Sigmoid	Log-Sigmoid
Parameters of sparsity	4	4
Weight sparsity	0.1	0.1
Decay value of weight	0.002	0.002
Iterations (max)	200	200
Final soft-max:		
Function for activation	Soft-max	Soft-max
Iteration (max)	200	200
Pre-training learning rate	0.000001	0.000001
The finer tuning learning rate	0.000001	0.000001
Fine-tune iteration (max)	100	100

The learning cost function was computed using the mean square error (MSE) function. Figures 11 and 12 show the learning curves for the proposed hybrid Gabor filter with the SSAE deep neural network model for the OLR and Extended Yale-B database.

Based on the two databases, two deep neural networks were trained for face image recognition: one using the proposed hybrid Gabor filter with the SSAE deep neural network model, and the other using a conventional SSAE deep neural network model. In order to evaluate the proposed method's performance at classifying new cases, the OLR database was tested with 80 samples of face images, while the Extended Yale-B database was tested with 78 samples of face images that were not used during the training session. Equation (12) was used to calculate the recognition rates:

$$(r.r) = \frac{number\ of\ face\ samples\ classified\ correctly}{Total\ number\ of\ test\ face\ samples} \tag{12}$$

Table 3 displays the computation time for the proposed hybrid Gabor filter with the SSAE method and the conventional SSAE network based on the OLR database. Table 4, on the other hand, shows the computational time of the proposed hybrid Gabor filter with the SSAE network and the conventional SSAE method based on the Extended Yale-B database.
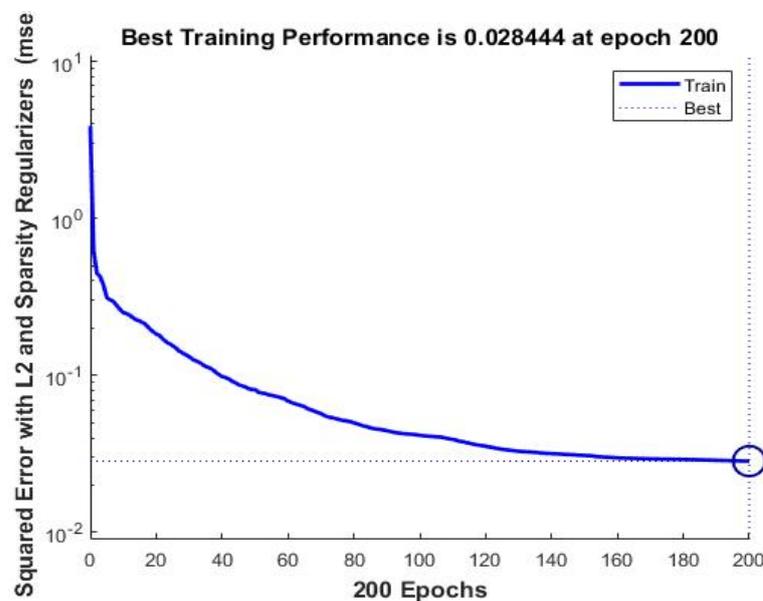
Figure 11. The learning curve for the proposed SSAE deep neural network for the OLR database.
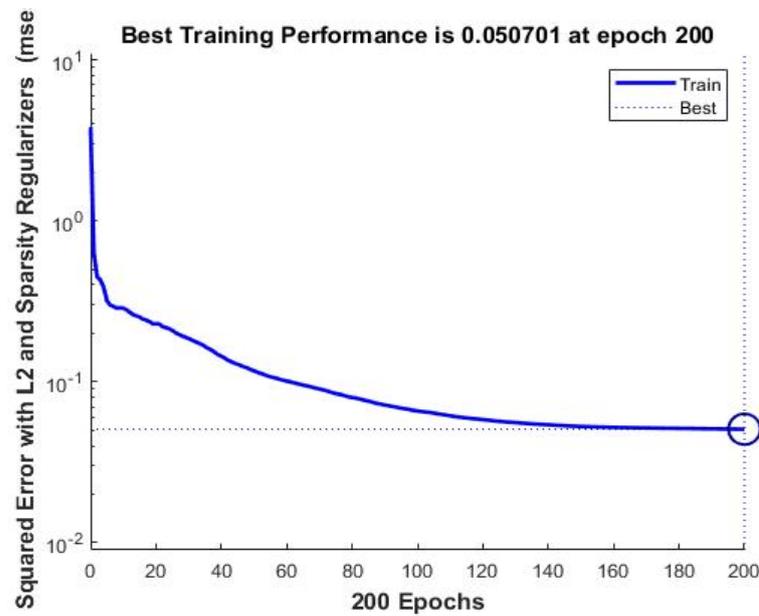
Figure 12. The learning curve for the proposed SSAE deep neural network for the Extended Yale-B database.

Table 3. Execution time computed for the OLR database.

SN of Images	Name of Images in the Database	Execution Time of the Hybrid Method	Execution Time of Conventional SSAE
1	01_OLR01	0.2973252	1.6227453
2	02_OLR02	0.2422379	0.4341096
3	03_OLR01	0.2412173	0.3973739
4	04_OLR02	0.2551962	0.2599288
5	05_OLR01	0.2518286	0.5701508
6	06_OLR02	0.2445681	0.2350624
7	08_OLR02	0.2546507	0.2208476
8	10_OLR01	0.2550419	0.2275963
9	12_OLR01	0.2448359	0.2136016
10	13_OLR02	0.2457450	0.2273580
11	15_OLR01	0.2505601	0.2269222
12	16_OLR02	0.2495898	0.2143724
13	17_OLR01	0.2435430	0.2196226
14	18_OLR02	0.2411174	0.2242768
15	19_OLR01	0.2409688	0.2173489
16	21_OLR02	0.2445355	0.2295809
17	23_OLR01	0.2492679	0.2278315
18	24_OLR02	0.2642704	0.2073777
19	25_OLR01	0.2477056	0.2160182
20	29_OLR01	0.2517592	0.2011851
21	31_OLR01	0.2459486	0.2240650
22	31_OLR02	0.2419242	0.2084263
23	32_OLR02	0.2494917	0.2226422
24	33_OLR01	0.2417259	0.2219115
25	34_OLR02	0.2469378	0.2150088
26	35_OLR01	0.2532889	0.2265672
27	36_OLR01	0.2419414	0.2074002
28	38_OLR01	0.2472856	0.2138572
29	39_OLR01	0.2515196	0.2170947
30	40_OLR01	0.2482124	0.2141638
Average execution time		0.2494747	0.2921483

Tables 5 and 6 show a performance comparison of face recognition efficiency between the proposed hybrid Gabor filter with the SSAE method and the conventional SSAE deep neural network method. The performance of the OLR database and the Extended Yale-B database were measured in terms of MSE, classification precision, and recognition rate.

Table 4. Execution time computed for the Extended Yale-B database.

SN of Images	Image ID in Database	Execution Time of Hybrid Method	Execution Time of Conventional SSAE
1	01_YB_01	1.0825900	0.5950821
2	02_YB_01	0.6231380	0.5850558
3	03_YB_02	0.6030670	0.5713341
4	04_YB_02	0.5553510	0.5623317
5	05_YB_02	0.7251700	0.5687663
6	06_YB_01	0.5211130	0.5678378
7	07_YB_01	0.5070610	0.5625726
8	08_YB_02	0.4905720	0.5635912
9	10_YB_01	0.5109780	0.5907825
10	10_YB_02	0.5028580	0.5717226
11	14_YB_01	0.4994620	0.5665855
12	15_YB_02	0.4910670	0.5693121
13	16_YB_01	0.5557580	0.5772575
14	18_YB_01	0.4915970	0.5866595
15	19_YB_01	0.4930950	0.5742290
16	20_YB_02	0.4936690	0.5866195
17	22_YB_02	0.5115960	0.5715895
18	23_YB_01	0.5485060	0.5794412
19	24_YB_02	0.5161060	0.5671572
20	25_YB_02	0.4984720	0.5666622
21	27_YB_02	0.5013040	0.5764131
22	28_YB_02	0.4901850	0.5709971
23	30_YB_02	0.4971250	0.5775362
24	31_YB_02	0.4892620	0.6135733
25	33_YB_02	0.5119080	0.5658698
26	35_YB_01	0.6732850	0.5691650
27	35_YB_02	0.6152030	0.5742719
28	36_YB_02	0.4926430	0.5595510
29	37_YB_01	0.5000210	0.5697548
30	38_YB_02	0.4935000	0.5753255
Average execution time		0.5495220	0.5745683

Table 5. Performance comparison for the OLR datasets.

Metrics	Proposed Hybrid Method	Conventional SSAE
Samples	80	80
Error Rate (MSE)	0.0000	0.0009
Perfectly recognized images	80	79
Recognition rate (%)	100%	98.75

Table 6. Performance comparison for the Extended Yale-B dataset.

Metrics	Proposed Hybrid Method	Conventional SSAE
Samples	76	76
Error Rate (MSE)	0.0000	0.0055
Perfectly recognized images	76	71
Recognition rate (%)	100%	93.4211

4. Discussion

Tables 3 and 4 compare the OLR and Extended Yale-B database execution times for the proposed hybrid Gabor Filter and the SSAE deep neural network, as well as the conventional SSAE method. According to Table 3, the proposed method takes less time to execute for the OLR datasets than the conventional SSAE model, with an average execution time of 0.2495 s for the proposed method and an average execution time of 0.2921 s for the traditional SSAE model. Furthermore, according to Table 4, the proposed method takes less time to execute than the conventional method of SSAE for the Extended Yale-B database, with an average execution time of 0.5495 s against the average execution time of 0.5746 s for conventional SSAE. Therefore, the proposed method of the hybrid Gabor filter and SSAE deep neural network is faster than the conventional SSAE method.

As shown in Tables 5 and 6, the MSE for the proposed hybrid method using the OLR and the Extended Yale-B databases are lower than that of the conventional SSAE model. Accordingly, from the two databases, the MSE value of the proposed hybrid method is 0.0000, whereas the MSE values of the traditional SSAE model are 0.0009 and 0.0055 for both the OLR and the Extended Yale-B datasets, respectively. Furthermore, Table 5 shows that the proposed hybrid method achieved higher recognition rates with the test dataset on the OLR datasets than the conventional SSAE method alone. The proposed hybrid method

is 100% accurate, whereas traditional SSAE is 98.75% accurate. Finally, the proposed hybrid method resulted in higher recognition rates on the test dataset from the Extended Yale-B database than the equivalent conventional SSAE model. The proposed hybrid method is 100% accurate, whereas conventional SSAE is 93.42% accurate.

Performance Comparison of the Proposed Hybrid Method with the Existing Face Recognition Methods

This section uses the OLR and the Extended Yale-B database as an accuracy metric to evaluate the proposed hybrid method of the Gabor filter and SSAE deep neural network performs against the state-of-the-art techniques of face recognition. Table 7 shows that the precision of the proposed hybrid method is comparable to the advanced strategies for the OLR database. In addition, Table 8 indicates that the accuracy of the proposed hybrid Gabor filter and SSAE method is the highest compared to state-of-the-art techniques for the Extended Yale-B database.

Table 7. Accuracy Analysis based on the OLR dataset.

Method	Accuracy
Kamencay et al. [30]	98.3%
Tan et al. [31]	74.6%
Rajeesh [32]	96.0%
Zafaruddin and Fadewar [7]	93.0%
Proposed hybrid method	100.0%

Table 8. Accuracy Analysis based on the Extended Yale-B dataset.

Method	Accuracy
Fernades and Bala [33]	97.50%
Cai et al. [14]	95.17%
Kumar et al. [34]	99.6%
Proposed hybrid method	100.0%

Figures 13 and 14 graphically compare the proposed hybrid method of Gabor filter and SSAE deep neural network to the selected state-of-the-art face recognition methods based on the OLR and the Extended Yale-B database. From these figures, it can be deduced that the proposed hybrid method outperforms its equivalent face recognition methods.
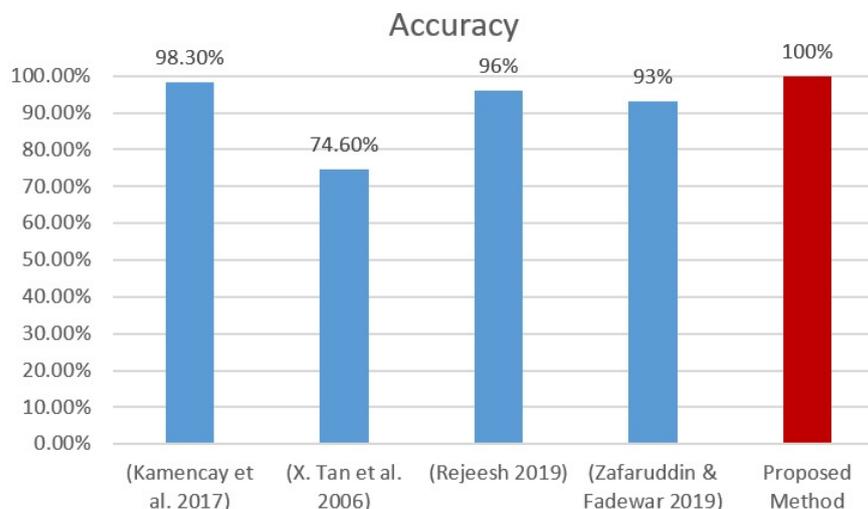
Figure 13. Accuracy analysis for OLR dataset [7,30–32].
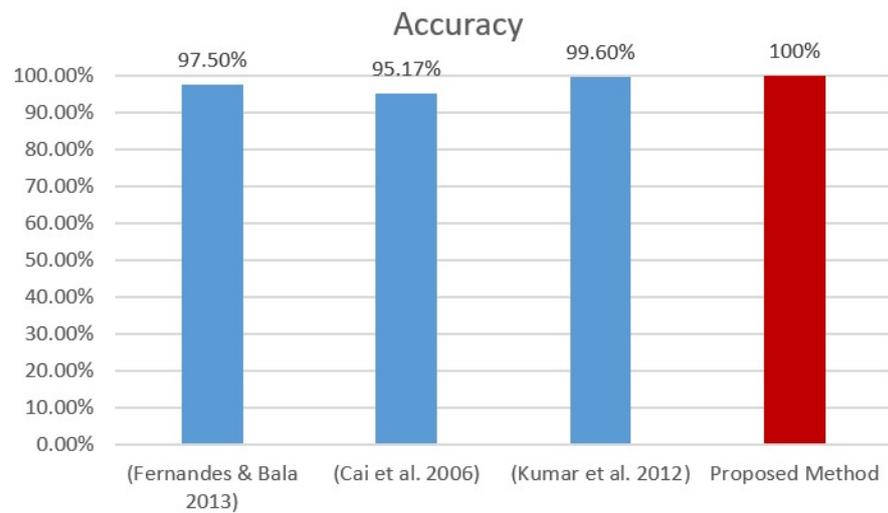
Figure 14. Accuracy analysis for Extended Yale-B database [14,33,34].

According to Tables 7 and 8, the proposed hybrid method outperformed the state-of-the-art equivalent methods in terms of recognition method accuracy metric for both the OLR and the Extended Yale-B databases. In summary, the proposed hybrid method achieved a 100% precision rate. The captured emotions-related face images were investigated, and the face has been recognized by applying the hybrid Gabor filter and SSAE method. The obtained classification accuracy results of the OLR and the Extended Yale-B datasets for different emotions are illustrated in Figures 15 and 16, respectively.
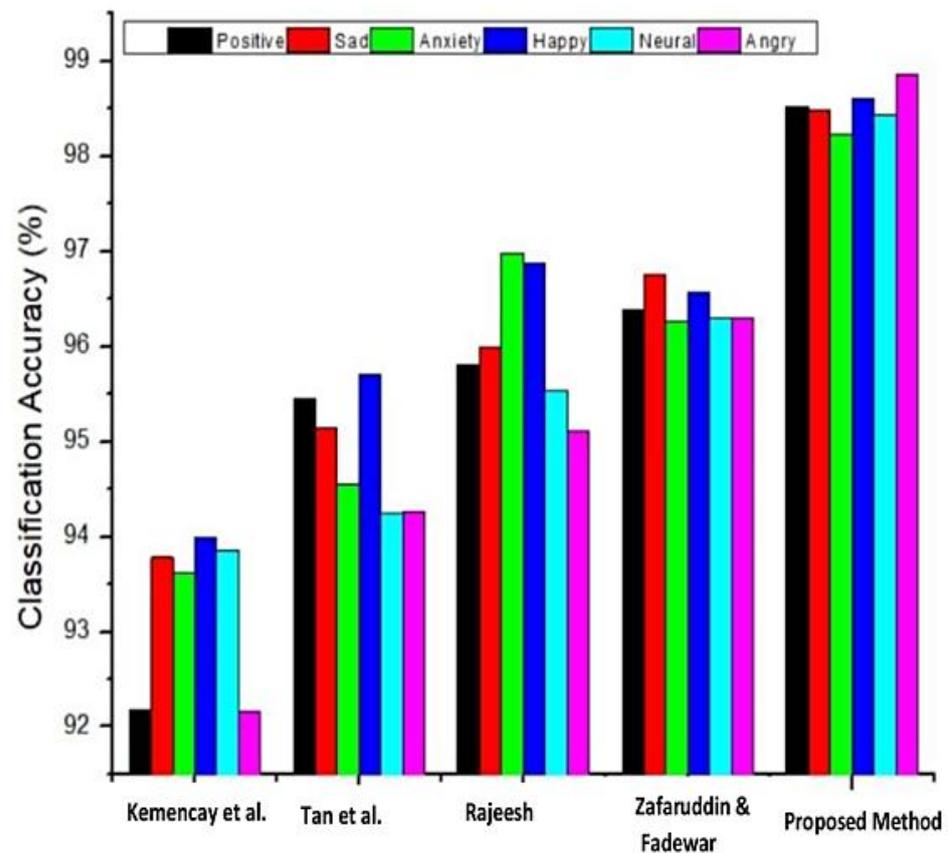
Figure 15. Classification accuracy analysis for different emotions based on the OLR dataset [7,30–32].
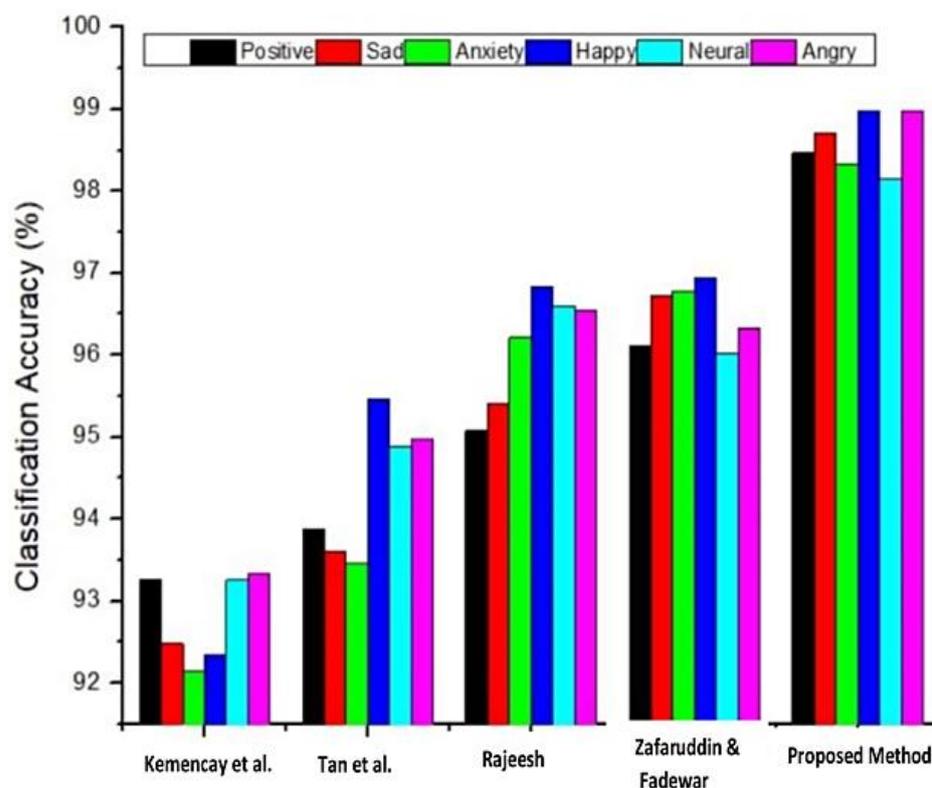
Figure 16. Classification accuracy analysis for different emotions base on the Extended Yale-B dataset [7,30–32].

Figures 15 and 16 denote the OLR and Extended Yale-B datasets classification accuracy using different face image emotions. The analysis indicates that the proposed hybrid SSAE approach attains 98.53% accuracy on the OLR dataset and 98.60% on the Extended Yale-B dataset when different face image emotions were used. The obtained results are maximum compared to other methods. On the contrary, the accuracy of other methods that were used for the comparison are thus summarized: Rajeesh [32] achieved 96.0%, Tan et al. [31] achieved 74.6%, Kamencay et al. [30] achieved 98.3%, and Zafaruddin and Fadewar [7] achieved 93.0% accuracies, respectively. With a 100% accuracy rate, the proposed hybrid method outperformed other face recognition methods with the Extended Yale-B database, according to Table 8. The accuracy of several other methods are, however, summarized as follows: Fernades and Bala [33] achieved 97.5%, Cai et al. [34] achieved 95.2%, and Kumar et al. [34] achieved 99.6%, respectively. Finally, for both the OLR and Extended Yale-B databases, the proposed hybrid method performed better than existing cutting-edge face recognition techniques.

5. Conclusions

This paper develops a novel hybrid method of face pattern recognition using a Gabor filter and an SSAE deep neural network. The proposed new face recognition system deals with feature extraction by comparing its performance to the conventional SSAE model and a few selected state-of-the-art methods. Furthermore, face image datasets from the OLR database and cropped versions of the Extended Yale-B database were used in all experiments in this study.

This paper describes a face recognition method that uses a hybrid Gabor filter and SSAE model. The Gabor filter feature extraction method was used to extract the initial face image features from the training datasets. The initial face image features were then fed into the SSAE network in order to reduce the extraction time required for face recognition due to different types of noise and deformations. Our findings support the proposed uniqueness and demonstrate that it is ideal for characterizing both basic and complicated

faces at the same time, regardless of the impact of other changes, such as scale, noise, and rotation. Finally, this study suggests that the proposed method be improved further in the future. Currently, the proposed system's extracted features have a large dimension while processing a large volume of data, resulting in a long processing time. It is therefore recommended that the computation complexity be reduced by incorporating the optimized feature selection approach to improve the overall face recognition process.

Author Contributions: Conceptualization, A.G.J.; methodology, A.G.J., R.C.M., O.L.U. and H.K.R.S.; software, A.G.J. and R.C.M.; validation, A.G.J., R.C.M., O.L.U. and H.K.R.S.; formal analysis, A.G.J., R.C.M., O.L.U. and H.K.R.S.; investigation, A.G.J.; resources, A.G.J., R.C.M., O.L.U. and H.K.R.S.; data curation, A.G.J.; writing—original draft preparation, A.G.J.; writing—review and editing, A.G.J., R.C.M., O.L.U. and H.K.R.S.; visualization, A.G.J.; supervision, R.C.M.; project administration, R.C.M.; funding acquisition, R.C.M. All authors have read and agreed to the published version of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are available at https://cam-orl.co.uk/facedatabase.html and http://vision.ucsd.edu/leekc/ExtYaleDatabase/ExtYaleB.html (accessed on 10 June 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Sanchez-Moreno, A.S.; Olivares-Mercado, J.; Hernandez-Suarez, A.; Toscano-Medina, K.; Sanchez-Perez, G.; Benitez-Garcia, G. Efficient Face Recognition System for Operating in Unconstrained Environments. *J. Imaging* **2021**, *7*, 161. [CrossRef] [PubMed]
2. Meshgini, S.; Aghagolzadeh, A.; Seyedarabi, H. Face recognition using Gabor-based direct linear discriminant analysis and support vector machine. *Comput. Electr. Eng.* **2013**, *39*, 727–745. [CrossRef]
3. Lu, D.; Yan, L. Face Detection and Recognition Algorithm in Digital Image Based on Computer Vision Sensor. *J. Sens.* **2021**, *2021*, 4796768. [CrossRef]
4. Reddy, A.H.; Kolli, K.; Kiran, Y.L. Deep cross feature adaptive network for facial emotion classification. *Signal Image Video Process.* **2021**, *16*, 369–376. [CrossRef]
5. Aldhahab, A.; Ibrahim, S.; Mikhael, W.B. Stacked Sparse Autoencoder and Softmax Classifier Framework to Classify MRI of Brain Tumor Images. *Int. J. Intell. Eng. Syst.* **2020**, *13*, 268–279. [CrossRef]
6. Görgel, P.; Simsek, A. Face recognition via Deep Stacked Denoising Sparse Autoencoders (DSDSA). *Appl. Math. Comput.* **2019**, *355*, 325–342. [CrossRef]
7. Zafaruddin, G.; Fadewar, H.S. Face Recognition Using Eigenfaces. In *Computing, Communication and Signal Processing*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 855–864.
8. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1097–1105. Available online: http://code.google.com/p/cuda-convnet/ (accessed on 21 April 2020).
9. Usman, O.L.; Muniyandi, R.C. CryptoDL: Predicting Dyslexia Biomarkers from Encrypted Neuroimaging Dataset Using Energy-Efficient Residue Number System and Deep Convolutional Neural Network. *Symmetry* **2020**, *12*, 836. [CrossRef]
10. Usman, O.L.; Muniyandi, R.C.; Omar, K.; Mohamad, M. Advance Machine Learning Methods for Dyslexia Biomarker Detection: A Review of Implementation Details and Challenges. *IEEE Access* **2021**, *9*, 36879–36897. [CrossRef]
11. Usman, O.L.; Muniyandi, R.C.; Omar, K.; Mohamad, M. Gaussian smoothing and modified histogram normalization methods to improve neural-biomarker interpretations for dyslexia classification mechanism. *PLoS ONE* **2021**, *16*, e0245579. [CrossRef]
12. Shen, L.L.; Bai, L.; Fairhurst, M. Gabor wavelets and General Discriminant Analysis for face identification and verification. *Image Vis. Comput.* **2007**, *25*, 553–563. [CrossRef]
13. Shen, L.; Bai, L. A review on Gabor wavelets for face recognition. *Pattern Anal. Appl.* **2006**, *9*, 273–292. [CrossRef]
14. Cai, D.; He, X.; Han, J.; Zhang, H.-J. Orthogonal Laplacian faces for 3D face recognition. *IEEE Trans. Image Process.* **2006**, *15*, 3608–3614. [CrossRef]
15. Daugman, J.G. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am. A* **1985**, *2*, 1160–1169. [CrossRef] [PubMed]
16. Hamamoto, Y.; Uchimura, S.; Watanabe, M.; Yasuda, T.; Mitani, Y.; Tomita, S. A gabor filter-based method for recognizing handwritten numerals. *Pattern Recognit.* **1998**, *31*, 395–400. [CrossRef]

17. Wang, X.; Ding, X.; Liu, C. Gabor filters-based feature extraction for character recognition. *Pattern Recognit.* **2005**, *38*, 369–379. [CrossRef]

18. Kamarainen, J.-K.; Kyrki, V.; Kalviainen, H. Invariance properties of Gabor filter-based features-overview and applications. *IEEE Trans. Image Process.* **2006**, *15*, 1088–1099. [CrossRef]

19. Liu, C.; Wechsler, H. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. *IEEE Trans. Image Process.* **2002**, *11*, 467–476. [CrossRef]

20. Jean Effil, N.; Rajeswari, R. Wavelet scattering transform and long short-term memory network-based noninvasive blood pressure estimation from photoplethysmograph signals. *Signal Image Video Process.* **2021**, *16*, 1–9. [CrossRef]

21. Rahman, M.A.; Muniyandi, R.C.; Albashish, D.; Rahman, M.M.; Usman, O.L. Artificial neural network with Taguchi method for robust classification model to improve classification accuracy of breast cancer. *PeerJ Comput. Sci.* **2021**, *7*, e344. [CrossRef]

22. Rahman, M.M.; Usman, O.L.; Muniyandi, R.C.; Sahran, S.; Mohamed, S.; Razak, R.A. A Review of Machine Learning Meth-ods of Feature Selection and Classification for Autism Spectrum Disorder. *Brain Sci.* **2020**, *10*, 949. [CrossRef]

23. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef] [PubMed]

24. Gideon, S. Estimating the Dimension of a Model Source. *Ann. Stat.* **2008**, *6*, 461–464.

25. Fuad, T.H.; Fime, A.A.; Sikder, D.; Iftee, A.R.; Rabbi, J.; Al-Rakhami, M.S.; Gumaei, A.; Sen, O.; Fuad, M.; Islam, N. Recent Advances in Deep Learning Techniques for Face Recognition. *IEEE Access* **2021**, *9*, 99112–99142. [CrossRef]

26. Ng, A. Sparse autoencoder. In *CS294A Lecture Notes*; Stanford University: Stanford, CA, USA, 2011; pp. 1–19.

27. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef]

28. Maroosi, A.; Muniyandi, R.C.; Sundararajan, E.; Zin, A.M. Parallel and distributed computing models on a graphics processing unit to accelerate simulation of membrane systems. *Simul. Model. Pract. Theory* **2014**, *47*, 60–78. [CrossRef]

29. Rahman, M.A.; Muniyandi, R.C. Review of GPU implementation to process of RNA sequence on cancer. *Inform. Med. Unlocked* **2018**, *10*, 17–26. [CrossRef]

30. Kamencay, P.; Benčo, M.; Miždoš, T.; Radil, R. A new method for face recognition using convolutional neural network. *Digit. Image Process. Comput. Graph.* **2017**, *16*, 663–672. [CrossRef]

31. Tan, X.; Chen, S.; Zhou, Z.-H.; Li, J. Learning Non-Metric Partial Similarity Based on Maximal Margin Criterion. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 17–22 June 2006; pp. 138–145. [CrossRef]

32. Rejeesh, M. Interest point based face recognition using adaptive neuro fuzzy inference system. *Multimed. Tools Appl.* **2019**, *78*, 22691–22710. [CrossRef]

33. Fernandes, S.; Bala, J. Performance Analysis of PCA-based and LDA-based Algorithms for Face Recognition. *Int. J. Signal Process. Syst.* **2013**, *1*, 1–6. [CrossRef]

34. Kumar, R.; Banerjee, A.; Vemuri, B.C.; Pfister, H. Trainable Convolution Filters and Their Application to Face Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *34*, 1423–1436. [CrossRef]