Hindawi
Journal of Robotics
Volume 2023, Article ID 9961421, 13 pages
https://doi.org/10.1155/2023/9961421

Research Article

A New Proposed Hybrid Learning Approach with Features for Extraction of Image Classification

Mohanad Azeez Joodi, Muna Hadi Saleh, and Dheyaa Jasim Kadhim

Department of Electrical Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq

Correspondence should be addressed to Mohanad Azeez Joodi; m.mohammad1702d@coeng.uobaghdad.edu.iq

Received 8 January 2023; Revised 21 March 2023; Accepted 31 March 2023; Published 15 April 2023

Academic Editor: Oscar Reinoso

Image classification is the process of finding common features in images from various classes and applying them to categorize and label them. The main problem of the image classification process is the abundance of images, the high complexity of the data, and the shortage of labeled data, presenting the key obstacles in image classification. The cornerstone of image classification is evaluating the convolutional features retrieved from deep learning models and training them with machine learning classifiers. This study proposes a new approach of "hybrid learning" by combining deep learning with machine learning for image classification based on convolutional feature extraction using the VGG-16 deep learning model and seven classifiers. A hybrid supervised learning system that takes advantage of rich intermediate features extracted from deep learning compared to traditional feature extraction to boost classification accuracy and parameters is suggested. They provide the same set of characteristics to discover and verify which classifier yields the best classification with our new proposed approach of "hybrid learning." To achieve this, the performance of classifiers was assessed depending on a genuine dataset that was taken by our camera system. The simulation results show that the support vector machine (SVM) has a mean square error of 0.011, a total accuracy ratio of 98.80%, and an $F1$ score of 0.99. Moreover, the results show that the LR classifier has a mean square error of 0.035 and a total ratio of 96.42%, and an $F1$ score of 0.96 comes in the second place. The ANN classifier has a mean square error of 0.047 and a total ratio of 95.23%, and an $F1$ score of 0.94 comes in the third place. Furthermore, RF, WKNN, DT, and NB with a mean square error and an $F1$ score advance to the next stage with accuracy ratios of 91.66%, 90.47%, 79.76%, and 75%, respectively. As a result, the main contribution is the enhancement of the classification performance parameters with images of varying brightness and clarity using the proposed hybrid learning approach.

1. Introduction

A hybrid supervised learning approach performs ethnicity categorization using both the strength of CNN and the rich features of the network. The technique combines the soft probability of CNN classification output with the engine of image ranking, utilizing the similarity between the query and the dataset pictures hierarchical characteristics. The combined feature vectors are trained using supervised support vector machine (SVM) hybrid learning to conduct ethnicity categorization [1]. Fuzzy removing redundancy restricted Boltzmann machine (F3RBM) is created to increase the efficiency of feature extraction and reduce learning time. For F3RBM-SVM model creation, the features generated by the F3RBM with unsupervised learning were imported into the support vector machine (SVM). This model provides quick and accurate automatic categorizations of various samples [2]. For deep characteristics extraction from photos of different faces, several deep convolutional neural networks (CNNs) were used. Support vector machine (SVM) and k-nearest neighbors (K-NN) are two examples of machine learning classifiers used for analyzing the further retrieved features. For the performance comparison of each model, several metrics, such as accuracy and precision, were employed and studied [3]. A comparison of using transfer learning with a classification technique, such as (support vector machines and modern pretrained CNN hybrid classifier, which is known as a support vector machine

(SVM), the Cohn-Kanade+ (CK+) database, and the natural visible and infrared expression (NVIE) database) two extremely well-liked expression databases, has been used for the testing. According to previous findings [4], pretrained CNNs generate superior results than customized techniques. Machine learning is a vital part of artificial intelligence (AI), which is essential for analyzing massive data sets (ML). The training dataset is used as input for supervised machine learning (SML), which frequently produces the desired output and makes predictions. Popular SML techniques used by academics include naive Bayes, logistic regression, random forest, J48, CART, Multilayer perception, and support vector machine (SVM). This study evaluates 305 publications on SML classification algorithms in its initial compilation using the systematic literature review (SLR) methodology [5, 6]. Several neural network classifiers are applied to the Twitter-collected dataset of people's faces. Human accuracy is just 26.96%, whereas the best accuracy attained is 53.2% [7, 8]. A linear SVM classifier and a multilabel, deep learning-based facial action detector beat cutting-edge methods (HOG and LBP), are used. By learning internal data structures, storing face motions, and giving a hierarchical representation of facial traits, this method can be applied to additional datasets [9]. In order to characterize erasures in child-written writings, a number of machine learning (ML) techniques are examined, including the support vector machine (SVM), boosted and bagged decision trees, K-nearest neighbor, Naive Bayes, discriminate analysis, logistic regression, and deep neural networks [10, 11]. The training classification uses local feature extraction and global feature extraction. Based on the deep learning algorithm, the data set preprocessed by the filter was fed into the network using the local quantization technique. Furthermore, the CNN network was prepared to create high-performing classification models [12, 13].

The main goal of this study is the classification using hybrid learning, which divides the dataset of face images into two categories: permission (negative class) (zero) and nonpermission (positive class) (one). This procedure is designed to identify those people who are permitted to enter high-security locations such as airports or tourist sites. Among deep learning and machine learning, one uses features (convolutional features) extracted from one of the deep learning models, such as the VGG-16 model. Moreover, rendering this model extracts features only without classifying them to get large numbers of features extracted from deep learning to boost classification accuracy and classification performance parameters when compared with traditional feature extraction methods such as scale invariant feature transform (SIFT) and speed up robust features (SURF) in terms of time consumption and a limited number of features. It requires defrosting the VGG-16 model's end layers to extract the feature vector alone in order to train and test them using supervised machine learning classifiers such as artificial neural networks (ANNs), decision trees (DT), logistic regression (LR), Naive Bayes (NB), random forests (RF), support vector machines (SVM), and weighted K-nearest neighbors (WKNN). Each of the aforementioned classifiers is given the same set of characteristics to discover the best classification. To accomplish this, the performance of the classifiers is evaluated and assessed using real datasets from our cameras, which were used to create the study's dataset.

The work of the present paper is structured as follows: the background for the face recognition system is provided by the assessment of the feature extraction from deep learning models and their training using machine learning classifiers in Section 1. Section 2 divides the proposed hybrid learning approach, which may be used in the present study, into two parts: supervised machine learning classifiers and evaluation classification parameters to achieve the overall proposed categorization of hybrid systems. Section 3 describes the simulation result component. Section 4 discusses the main results and contributions.

2. Proposed Hybrid Learning Approach

Figure 1 represents the hybrid learning method, which consists of two stages: the first stage is to insert the real data set images created from our cameras into the Haar cascade detector or MTCNN detector to detect the faces and then into the VGG-16 deep learning model to extract the features (convolutional features) from them by making this model work without classifiers (defrosting the end layers of the model) [14]. The second stage is to input the features of each of the seven classifiers, such as the artificial neural network (ANN), decision tree (DT), logistic regression (LR), Naive Bayes (NB), random forest (RF), support vector machine (SVM), and weighted K-nearest neighbors (WKNN). To classify the decision into permission and nonpermission classes, it is crucial to pay attention to the result and at what time it uses the feature training among the hybrid deep learning and machine learning algorithms.

Table 1 represents the architecture of the VGG-16 deep learning model used to extract all features from 263 genuine dataset images with 150×150 pixels, where the number of features is 8192. The number of features extracted from deep learning is too large to be obtained by using machine learning techniques to boost classification accuracy and classification performance parameters. The kernel convolution filters were utilized in the VGG-16 model to extract the convolution features, such as identity filter, edge detection filter, sharpen filter, box blur (normalized) filter, and Gaussian blur (approximation) [15].

The real dataset, which includes faces with a minimum size of 150×150 pixels, utilized 263 photos with frontal and nonfrontal faces of various resolutions and brightness levels which were chosen from this collection after the Haar cascade detector or MTCNN detector and then categorized into the permission and nonpermission categories to identify those people who are permitted to enter high-security locations, such as airports or tourist sites.

In order to accomplish the suggested overall hybrid system categorization, the result component of this study is divided into two parts. The usage of machine learning classifiers is described in the first part. While in the second part, the application of a few classification performance

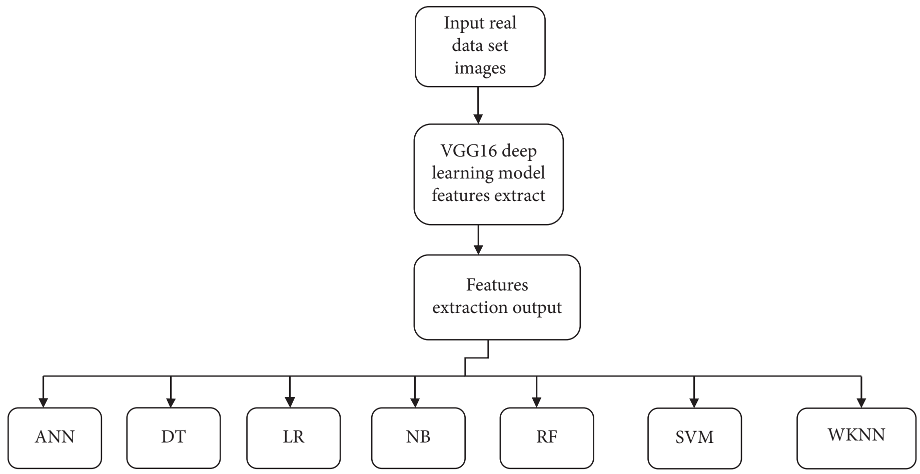

Figure 1: New proposed hybrid learning approach.

Table 1: VGG-16 architecture model.

Layer (type)	Output (shape)	Parameters
Input_1 (input layer)	(None, 150, 150, 3)	0
Block1_conv1 (Conv2D)	(None, 150, 150, 64)	1792
Block1_conv2 (Conv2D)	(None, 150, 150, 64)	36928
Block1_pool (MaxPooling2D)	(None, 75, 75, 64)	0
Block2_conv1 (Conv2D)	(None, 75, 75, 128)	73856
Block2_conv2 (Conv2D)	(None, 75, 75, 128)	147584
Block2_pool (MaxPooling2D)	(None, 37, 37, 128)	0
Block3_conv1 (Conv2D)	(None, 37, 37, 256)	295168
Block3_conv2 (Conv2D)	(None, 37, 37, 256)	590080
Block3_conv3 (Conv2D)	(None, 37, 37, 256)	590080
Block3_pool (MaxPooling2D)	(None, 18, 18, 256)	0
Block4_conv1 (Conv2D)	(None, 18, 18, 512)	1180160
Block4_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
Block4_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
Block4_pool (MaxPooling2D)	(None, 9, 9, 512)	0
Block5_conv1 (Conv2D)	(None, 9, 9, 512)	2359808
Block5_conv2 (Conv2D)	(None, 9, 9, 512)	2359808
Block5_conv3 (Conv2D)	(None, 9, 9, 512)	2359808
Block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0

metrics is explained. Examples of photos from the real dataset are shown in Figures 2 and 3.

2.1. Supervised Machine Learning Classifiers. This section describes seven classifiers, including the artificial neural network (ANN), decision tree (DT), logistic regression (LR), Naive Bayes (NB), random forest (RF), support vector machine (SVM), and weighted k-nearest neighbors (WKNN), which are used to train the features of the images in the dataset.

2.1.1. Artificial Neural Network Classifier (ANN). An artificial neural network typically has layers based on the structure of a human neuron. Layers are made up of several connected "nodes" with individual "activation functions." Possible layers of a neural network include the input layer, the hidden layer, and the output layer. The human brain has millions of neurons. It transmits, receives, and processes electrical and chemical signals. The synapses, a special structure, connect these neurons. Synapses allow neurons to communicate with one another. Huge numbers of simulated neurons form neural networks.

Although neural networks work well with both linear and nonlinear data, they require a wide range of training data to function in the real world, particularly in robots. This sounds believable because any machine that is being learned must have access to a sufficient number of representative examples in order to completely comprehend the underlying structure and generalize new circumstances. ANNs are simple mathematical models that can enhance existing data processing techniques. Although it falls short of the human brain's capabilities, it nonetheless constitutes the core of artificial intelligence [16].

ANNs aim to categorize an observation as belonging to a discrete class. Although the input characteristics (independent variables) may be of the categorical or numerical type, the dependent variable must be of the category type. After extracting features from the deep learning VGG-16 model, we input the features into an ANN classifier to train it and then test the images to determine the classification parameters (precision, recall (the true positive rate), test accuracy, training error, and $F1$ score).

2.1.2. Decision Trees Classifier (J48). The highly evolved and sophisticated version of the C4.5 algorithm is J48. A fresh sample from the tested dataset will be categorized by a decision tree based on the tree from the training dataset. The feature that allows for quicker identification of various samples depends on the amount of time spent on the training set. The anxious branch is a clear indication that any potential feature relevance should be discounted [17]. This classifier is considered the best among the six methods due to its widespread use and construction on information entropy [18]. With this approach, each aspect of the data might be used to break it down into smaller components, such as tree root nodes. Essentially, the leaves, internal node, and root node are the three nodes that comprise the entire

Figure 2: Faces with permission class.

Figure 3: Faces with nonpermission class.

tree anatomy. Since there are no incoming edges at the root node in this method, the leaf node is used to calculate the class label. J48 guarantees positive outcomes from all the created tree edges.

Classification trees are tree models with a limited number of possible values for the object variable. The branches of this tree structure reflect the combinations of qualities that reproduce certain class labels, while the leaves represent class labels. Decision trees may be created significantly quicker than previous classification methods [19].

Computer scientists, cognitive scientists, data miners, statisticians, biologists, and engineers frequently employ the notion of information theory. Entropy, a concept from information theory, quantifies the degree of uncertainty across controllable variables in a dataset. The entropy of a random variable is a widely understood idea. The formula reflects the information theory of entropy measurement (1). If X is an attribute, p is each element, and j is the location of each element of X; equation (1) is used to evaluate the entropy approach.

$$H(X) = \sum_{j=1}^{k} p_j \log_2 \frac{1}{p_j} = -\sum_{j=1}^{k} p_j \log_2 p_j. \tag{1}$$

Greater value $H(X)$ denotes a more random property of X. A property with a lower $H(X)$ value, on the other hand, indicates less randomness. After extracting features from the deep learning VGG-16 model, we input the features into a decision tree classifier to train it and then test the images to determine the classification parameters (precision, recall (the true positive rate), test accuracy, training error, and $F1$ score).

2.1.3. Logistic Regression. Logistic regression is a different statistical technique that machine learning has embraced. It is the strategy of preference for dealing with binary classification difficulties (problems with two classes of values). Statistics experts have developed the logistic function, often referred to as the sigmoid function, to describe the features of population growth in ecology, such as expanding quickly and peaking at the carrying capacity of the environment. The approach gets its name from its purpose. Using this S-shaped curve, every real-valued number may be changed into a value between 0 and 1, but never precisely in those ranges [20].

A mathematical function with a distinctive "S"-shaped curve, also known as sigmoid curve, is called a sigmoid function, as given in equation (2). Figure 4 shows the sigmoid function curve.

$$S(x) = \frac{1}{1 + e^{-x}}. \quad (2)$$

After extracting features from the deep learning VGG-16 model, we input the features into a logistic regression classifier to train it and then test the images to determine the classification parameters (precision, recall (the true positive rate), test accuracy, training error, and $F1$ score).

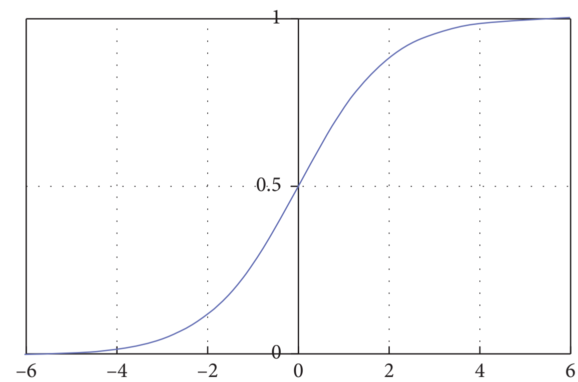

Figure 4: Sigmoid function curve [20].

2.1.4. Naive Bayes (NB). Naive Bayes classifiers are a class of classification algorithms based on the Bayes theorem. It is a set of algorithms rather than a single method, and they are all based on the assumption that each pair of characteristics being classified is independent of the other. Strong or naive independence between the properties of data points is an assumption made by Naive Bayes classifiers. Spam filters, text analysis, and medical diagnosis are some examples of common applications for Nave Bayes classifiers. Because it relies on the Bayes theorem concept, it is known as the Bayes principle [21].

NB is quick and easy to use due to its straightforward, efficient structure. Due to the separate estimation of each feature's likelihood, it is also helpful for large and dimensional data.

If C stands for the observation in X's class, the highest posterior probability of applying the Bayes rule to forecast the class of the observation X is shown in the following equation:

$$P(C/X) = \frac{P(C)P(X/C)}{P(X)}. \quad (3)$$

Using the premise whose features $X1$, $X2$, and Xn are conditionally independent of one another giving the class as shown in the following equation,

$$P(C\backslash X) = \frac{P(C)\prod_{i=1}^{n} P(X_i/C)}{P(X)}. \quad (4)$$

Equation (4) is adequate in addressing classification issues to forecast the most likely class giving a test observation. Figure 5 shows the Naïve Bayes principle.

In the formula, $i = 1, \ldots, n$ is used to estimate class probabilities $P(C)$ and conditional probabilities $P(X_i \mid C)$ as in (4) [22].

After extracting features from the deep learning VGG-16 model, we input the features into a Naive Bayes classifier to train it and then test the images to determine the classification parameters (precision, recall (the true positive rate), test accuracy, training error, and $F1$ score).

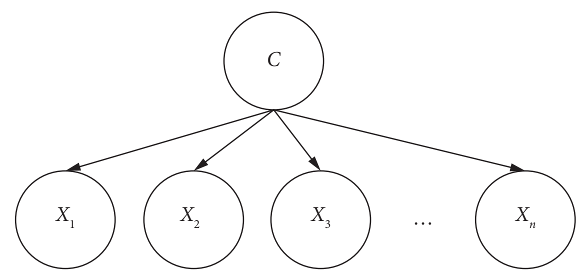

Figure 5: Naïve Bayes principle [22].

2.1.5. Random Forest (RF). A supervised computer called a random forest classifier creates and combines several decision trees to make a forest, which is how it works. By using this technique, operators may benefit from the use of various learning models to elevate accuracy to a new level. This distinctive difference in the method from other learning machines is the way root nodes are linked, which is redundant [23].

Classification is regarded as the foundation of machine learning. Figure 6 shows a random forest with two trees.

In a random forest, only random subsets of the qualities are considered when splitting a node. The randomization of the trees may be boosted by adding random thresholds for each feature in addition to the best available thresholds [24].

After extracting features from the deep learning VGG-16 model, we input the features into a random forest classifier to train it and then test the images to determine the classification parameters (precision, recall (the true positive rate), test accuracy, training error, and $F1$ score).

2.1.6. Support Vector Machine Classifier (SVM). Academics have taken notice of this approach because of its superb compatibility with machine learning applications involving enormous amounts of data, such as computer vision and pattern recognition [25]. This widely used classifier tries to create the best hyperplane-like margins, as illustrated in Figure 7. The goal of the support vector machine is to maximize the separation between the nearest training SAT samples and the hyperplane. According to various studies [26], the ideal hyperplane would offer greater accuracy with all types of information in a linearly divided environment.

After extracting features from the deep learning VGG-16 model, we input the features into a support vector machine classifier to train it and then test the images to determine the classification parameters (precision, recall (the true positive rate), test accuracy, training error, and $F1$ score).

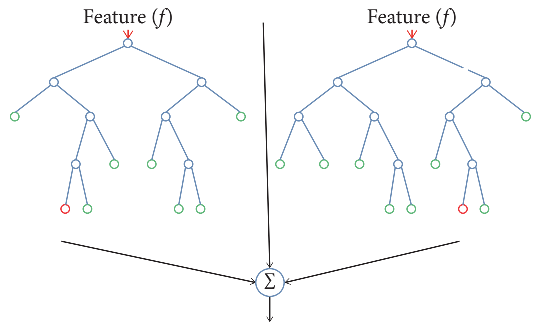

Figure 6: Random forest of two trees [24].

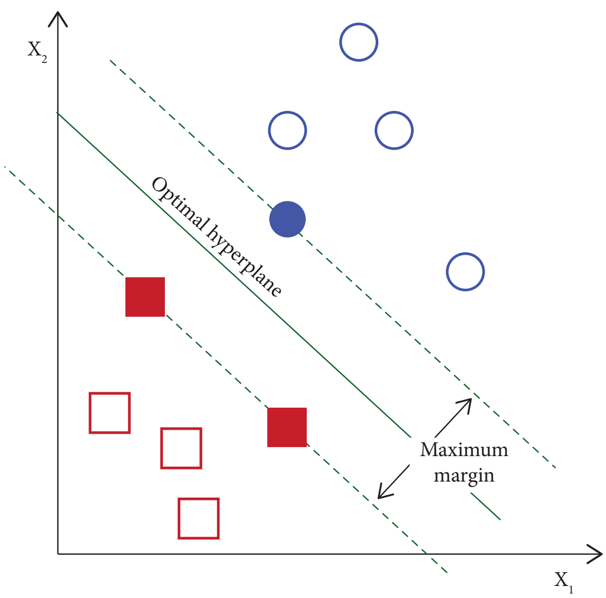

Figure 7: SVM hyperplane [26].

2.1.7. Weighted K-Nearest Neighbor Classifier (WKNN). Weighted K-NN is a version of “K-nearest neighbors.” The selection of the hyperparameter k is one of several factors that determine the performance of the K-NN algorithm. The strategy would be more susceptible to outliers if k was too small. There may be an excessive number of points from other classes nearby if k is too large.

The most straightforward approach is to conduct a majority vote. Although this might present issues if the closest neighbors are dispersed, clearly identifying the object class. The output is class-labeled, and K is usually a digit that is next to another positive integer. Based on the votes of the neighbors, a sample is classified according to the nature and position of the feature. To put it another way, if a sample demonstrates specific traits of a class while taking its separation from the class into consideration, it will be placed in that category [27].

The red markings represent class 0 points, whereas the green labels represent class 1 points. Figure 8 depicts the white point as the “enquiring point” (the point whose class label must be anticipated).

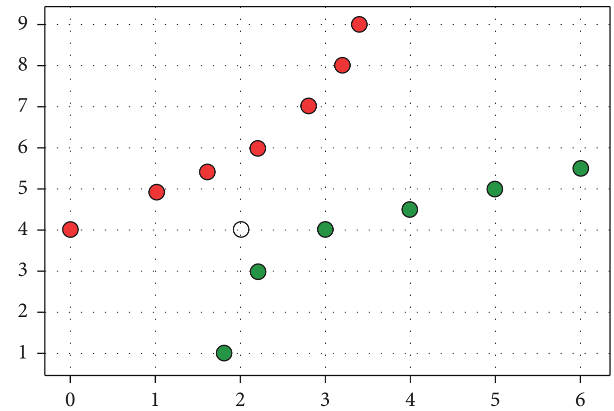

Figure 8: The white point serves as the inquiry point, while the red labels denote the class 0 points and the green labels the class 1 points [27].

A KNN-based classifier would assert that the query point belongs to class 0 if the aforementioned dataset was fed into it.

However, it is obvious from the plot that the point is closer to the class 1 points than the class 0 points. Weighted kNN is employed to get over this drawback.

In weighted KNN, the kernel function is used to assign a weight to the closest k points. In a weighted kNN, points that are closer to each other are heavy, and those that are further away are not heavy. Any function can be used as the kernel function for the weighted KNN classifier, whose value decreases as the distance increases. A simple function is the inverse distance function. WKNN algorithm steps are as follows:

(i) Let x represent a brand-new observation (query point) for which a class label prediction is required.

(ii) Assume that the training set $L = (x_i, y_i)$ consists of observations for the given class y_i. Determine the Euclidian distance Ed (x_i, y_i) for $(i = 1, \ldots, n)$ between each point in the training set and the query point (5).

$$\mathrm{Ed}(x, y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}. \tag{5}$$

(iii) Decide which of the k training data points, designated D’D, are the closest to the query points.

(iv) Determine the class of the query point using distance-weighted voting. The class labels are represented by the v in the following equation:

$$y' = \underset{v}{\operatorname{argmax}} \sum_{(x_i, y_i) \in D_z} w_i x I(v = y_i). \tag{6}$$

After extracting, we input the features into a weighted K-nearest neighbor classifier to train it and then test the images to determine the classification parameters (precision, recall (the true positive rate), test accuracy, training error, and $F1$ score).

2.2. Evaluation Classification Parameters. The following metrics were used to assess the performance of the proposed system: precision, recall (the true positive rate), test accuracy, training error, and $F1$ score as described in (7)–(13) expressions:

$$\text{precision} = \frac{tp}{tp + fp}, \tag{7}$$

$$\text{recall} = \frac{tp}{tp + fn}, \tag{8}$$

$$\text{test acc} = \frac{(tp + tn)}{(tp + fp + tn + fn)}, \tag{9}$$

$$\text{training error} = \frac{(fp + fn)}{(tp + fp + tn + fn)}, \tag{10}$$

$$\text{specificity} = \frac{tn}{tn + fp}, \tag{11}$$

$$\begin{aligned}\text{FPR (false positive rate)} &= 1 - \text{specificity}\\ &= \frac{fp}{tn + fp},\end{aligned} \tag{12}$$

$$F1 = \left(\frac{2 * (\text{precision} * \text{recall})}{(\text{precision} + \text{recall})}\right). \tag{13}$$

3. Results and Discussion

The confusion matrix result is shown in this section, along with the classification table for each machine classifier and the comparison of the simulation results between testing the images by seven types of machine classifiers after evaluating 84 support photos.

In the ANN classifier, Figure 9 shows the confusion matrix between the true label and predicted label. Table 2 shows the classification performance parameters.

In the DT classifier, Figure 10 shows the confusion matrix between the true label and predicted label. Table 3 shows the classification performance parameters.

In the logistic regression classifier, Figure 11 shows the confusion matrix between the true label and predicted label. Table 4 shows the classification performance parameters.

In the Naïve Bayes classifier, Figure 12 shows the confusion matrix between the true label and predicted label. Table 5 shows the classification performance parameters.

In the random forest classifier, Figure 13 shows the confusion matrix between the true label and predicted label. Table 6 shows the classification performance parameters.

In the support vector machine classifier, Figure 14 shows the confusion matrix between the true label and predicted label. Table 7 shows the classification performance parameters.

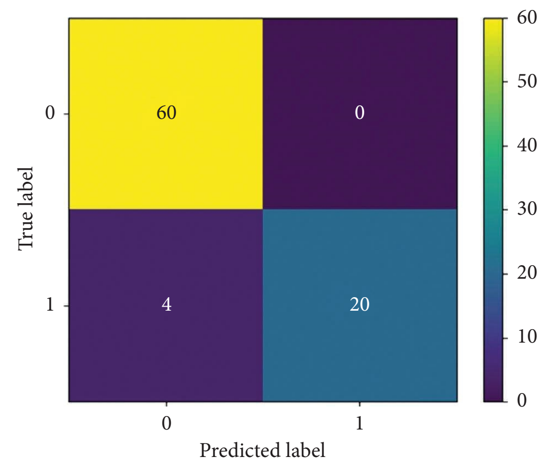

Figure 9: ANN classification confusion matrix.

In the WKNN classifier, Figure 15 shows the confusion matrix between the true label and predicted label. Table 8 shows the classification performance parameters.

Table 9 displays testing classification parameters such as precision, recall, FPR, $F1$ score, test accuracy, and mean square error for each instance of each classifier in relation to the feature selection approaches used in deep learning VGG-16 feature extraction training in seven supervised machine learning classifiers. In order to identify which classifier is the best, this approach compares them side by side.

The findings of this study show that support vector machines, with a total ratio of 98.80% and an $F1$ score of 0.99, outperform all other tested classifiers when the features (convolution features) from the VGG-16 deep learning model are used. LR has a total ratio of 96.42%, the $F1$ score of 0.96 comes in the second place, ANN has a total ratio of 95.23%, and the $F1$ score of 0.94 comes in the third place. Additionally, with their respective accuracy ratios of 91.66%, 90.47%, 79.76%, and 75%, RF, WKNN, DT, and NB with an $F1$ score go to the next stage, as shown in Table 9.

Figure 16 depicts the test accuracy of features trained by seven machine-learning classifiers with a mean square error. The support vector machine has a minimum mean square error of 0.011; the LR classifier has a mean square error of 0.035; the ANN classifier has a mean square error of 0.047; the RF classifier has a mean square error of 0.083; and the WKNN classifier has a mean square error of 0.0952. Additionally, with their respective mean square errors of 0.2023, 0.25, DT, and NB, they go to the next stage.

Table 10 compares the simulation results of our new proposed hybrid learning approach, which shows that the hybrid VGG-16 with the SVM, with LR, and with ANN machine classifiers has better classification metrics, such as test accuracy, precision, and recall, than the other approaches.

Figure 17 depicts the precision of our new proposed hybrid learning approach, VGG-16 with SVM hybrid machine learning, LR hybrid machine learning, and ANN hybrid machine learning, which has better precision of the

Table 2: ANN classifier parameters.

Precision	Recall (TPR)	FPR	$F1$ score	Test accuracy (%)	Mean square error	Support images
0.97	0.92	0	0.94	95.23	0.0476	84

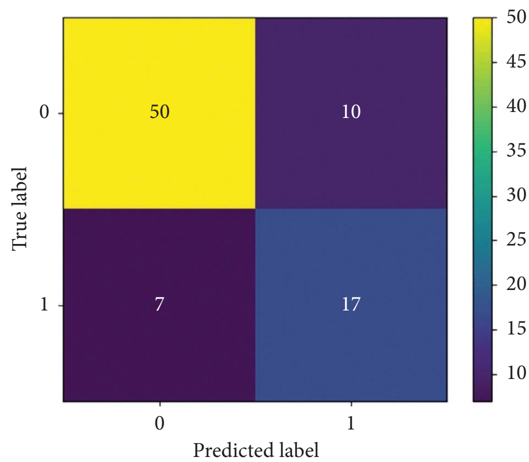

Figure 10: J48 classification confusion matrix.

Table 3: J48 classifier parameters.

Precision	Recall (TPR)	FPR	$F1$ score	Test accuracy (%)	Mean square error	Support images
0.75	0.77	0.370	0.76	79.76	0.2023	84

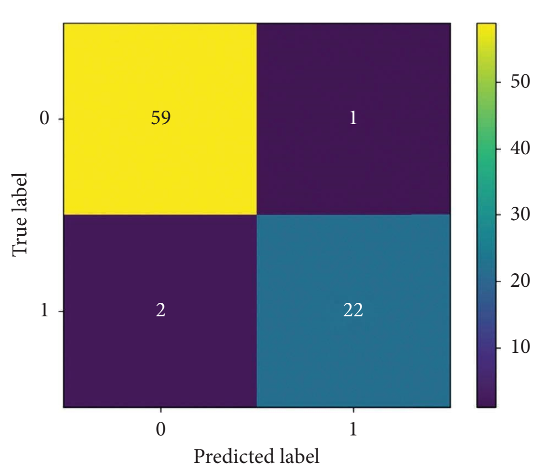

Figure 11: LR classification confusion matrix.

Table 4: LR classifier parameters.

Precision	Recall (TPR)	FPR	$F1$ score	Test accuracy (%)	Mean square error	Support images
0.96	0.95	0.043	0.96	96.42	0.035	84

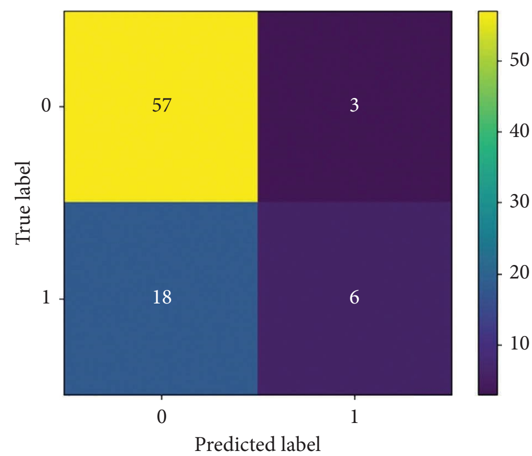

FIGURE 12: Naive Bayes classification confusion matrix.

TABLE 5: NB classifier parameters.

Precision	Recall (TPR)	FPR	$F1$ score	Test accuracy (%)	Mean square error	Support images
0.71	0.6	0.333	0.6	75	0.25	84

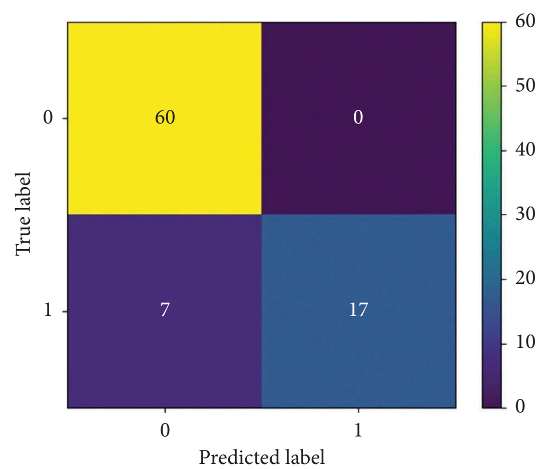

FIGURE 13: Random forest classification confusion matrix.

TABLE 6: RF classifier parameters.

Precision	Recall (TPR)	FPR	$F1$ score	Test accuracy (%)	Mean square error	Support images
0.95	0.85	0	0.89	91.66	0.083	84

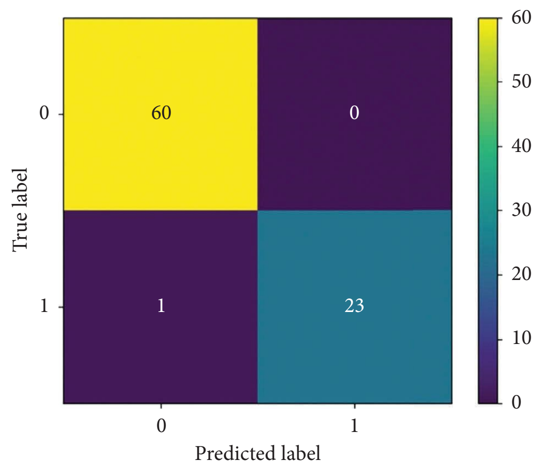

FIGURE 14: Support vector machine classification confusion matrix.

TABLE 7: SVM classifier parameters.

Precision	Recall (TPR)	FPR	$F1$ score	Test accuracy (%)	Mean square error	Support images
0.99	0.98	0	0.99	98.80	0.012	84

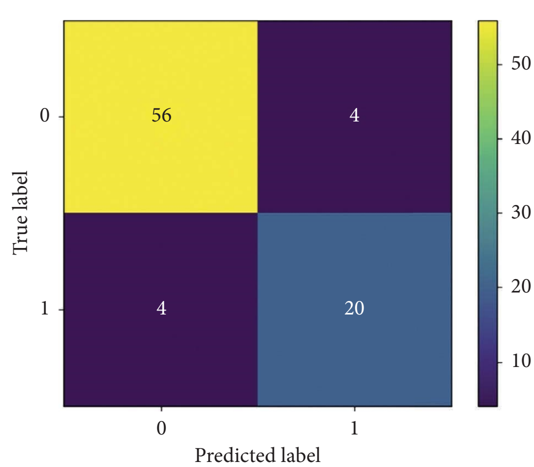

FIGURE 15: K-nearest neighbor classification confusion matrix.

TABLE 8: WKNN classifier parameters.

Precision	Recall (TPR)	FPR	$F1$ score	Test accuracy (%)	Mean square error	Support images
0.88	0.88	0.1666	0.88	90.47	0.0952	84

TABLE 9: Image testing in seven machine learning classifiers.

Classifier types	Precision	Recall (TPR)	FPR	*F*1 score	Test accuracy (%)	Mean square error	Support images
ANN	0.97	0.92	0	0.94	95.23	0.0476	84
DT	0.75	0.77	0.370	0.76	79.76	0.202	84
LR	0.96	0.95	0.043	0.96	96.42	0.035	84
NB	0.71	0.6	0.333	0.6	75	0.250	84
RF	0.95	0.85	0	0.89	91.66	0.083	84
SVM	0.99	0.98	0	0.99	98.80	0.012	84
WKNN	0.88	0.88	0.166	0.88	90.47	0.095	84

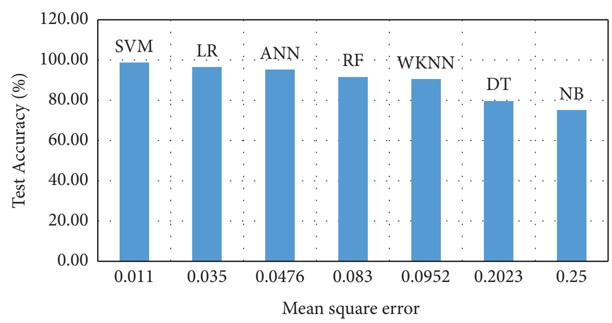

FIGURE 16: Depicting the test accuracy of images by seven machine learning classifiers with a mean square error.

TABLE 10: Comparison of the best new proposed hybrid learning approach.

New proposed hybrid learning approach	Precision	Recall (TPR)	Test accuracy (%)	Mean square error
VGG-16 with ANN machine classifier	0.97	0.92	95.23	0.0476
VGG-16 with LR machine classifier	0.96	0.95	96.42	0.035
VGG-16 with SVM machine classifier	0.99	0.98	98.80	0.011

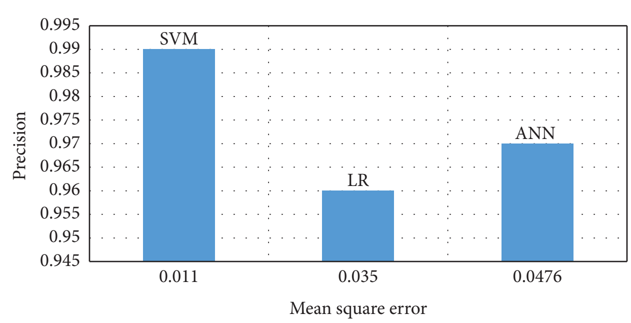

FIGURE 17: Depicting the precision of test images of the new proposed hybrid learning approach by SVM, LR, and ANN machine classifiers with mean square error.

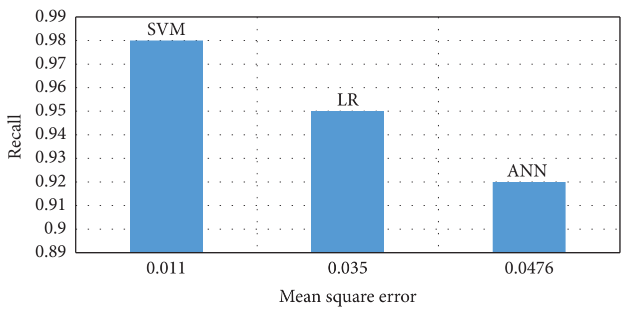

Figure 18: Depicting recall of test images of a new proposed hybrid learning approach by SVM, LR, and ANN machine classifiers with mean square error.

three hybrid approaches SVM, ANN, and LR than the other hybrid approaches.

Figure 18 depicts the recall of our new proposed hybrid learning approach, SVM, LR, and ANN machine classifiers with VGG-16 deep learning extract features, which have better recall of the three types SVM, LR, and ANN hybrid machine learning classifiers than the other hybrid approaches.

4. Conclusion

This work combines both deep learning and machine learning to recognize faces by extracting convolutional features from a deep learning model, such as the VGG-16 model, without classifying them to get large numbers of features extracted (8192 features) to boost classification accuracy and classification performance parameters first. Then, artificial neural networks (ANNs), decision trees (DT), logistic regression (LR), Naïve Bayes (NB), random forests (RF), support vector machines (SVM), and the weighted k-nearest neighbor classifier (WKNN) are examples of supervised machine learning classifiers that are used for training and evaluating. To do this, the performances of classifiers are evaluated and assessed. The dataset for the study was produced using actual data from our cameras. The experimental results show that support vector machines outperform all other tested classifiers when the features from the VGG-16 deep learning model are used, with a total test accuracy ratio of 98.80% and a mean square error of 0.011 as applied to the best high-ranked feature vector in seven classifiers. Currently, LR's total accuracy ratio and ANN's total accuracy ratio are 96.42% and 95.23%, respectively. Additionally, RF, WKN, DT, and NB advance to the following level with respective accuracy ratios of 91.66%, 90.47%, 79.76%, and 75%. In future work, other deep learning models to extract the features will be suggested, such as Res Net-50, Alex Net, and Google Net Inspection v3, with the same supervised machine learning classifiers, and the results will be compared with the proposed hybrid learning approach.

Data Availability

Data sharing is not applicable to this article as no datasets were generated in this research.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was funded by the University of Baghdad.

References

[1] Z. Heng, M. Dipu, and K.-H. Yap, "Hybrid supervised deep learning for ethnicity classification using face images," in *Proceedings of the 2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, IEEE, Florence, Italy, May 2018.

[2] X. Lü, L. Meng, C. Chen, and P. Wang, "Fuzzy removing redundancy restricted Boltzmann machine: improving learning speed and classification accuracy," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 10, pp. 1–2509, 2019.

[3] A. Oumina, N. El Makhfi, and M. Hamdi, "Control the covid-19 pandemic: face mask detection using transfer learning," in *Proceedings of the 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*, pp. 1–5, IEEE, Kenitra, Morocco, December 2020.

[4] S. Shaees, H. Naeem, M. Arslan, M. R. Naeem, S. H. Ali, and H. Aldabbas, "Facial emotion recognition using transfer learning," in *Proceedings of the 2020 International Conference on Computing and Information Technology (ICCIT-1441)*, pp. 1–5, IEEE, Tabuk, Saudi Arabia, September 2020.

[5] N. A. D. Suhaimi and H. Abas, "A systematic literature review on supervised machine learning algorithms," *Perintis Ejournal*, vol. 10, no. 1, pp. 1–24, 2020.

[6] S. Q. Jabbar, D. J. Kadhim, and Y. Li, "Developing a video buffer framework for video streaming in cellular networks," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6584845, 13 pages, 2018.

[7] V.-D. Nguyen, M. Tran, and J. Luo, "Exploring facial differences in European countries boundary by fine-tuned neural networks," in *Proceedings of the 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 268–271, IEEE, Miami, FL, USA, April 2018.

[8] M. A. Joodi, M. H. Saleh, and D. J. Kadhim, "Increasing validation accuracy of a face mask detection by new deep learning model-based classification," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 29, no. 1, pp. 304–314, 2023.

[9] A. Ghasemi, S. Denman, S. Sridharan, and C. Fookes, "Discovery of facial motions using deep machine perception," in *Proceedings of the 2016 IEEE winter conference on applications of computer vision (WACV)*, pp. 1–7, IEEE, NY, USA, March 2016.

[10] M. Tenório, E. Costa, and T. Vieira, "Erasure detection and recognition on children's manuscript essays," in *Proceedings of the 2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 670–676, IEEE, Boston, MA, USA, November 2017.

[11] S. Q. Jabbar and D. J. Kadhim, "A proposed adaptive bitrate scheme based on bandwidth prediction algorithm for smoothly video streaming," *Journal of Engineering*, vol. 27, no. 1, pp. 112–129, 2021.

[12] Q. Kuang, "Face image feature extraction based on deep learning algorithm," in *Journal of Physics: Conference*

*Series,*vol. 1852, no. 3, IOP Publishing, Article ID 032040, 2021.
[13] D. J. Kadhim, M. H. Saleh, and S. J. Abou-Loukh, "Evaluation of massive MIMO communication performance under a proposed improved MMSE," *Presented at the precoding (PIMPIAES International Journal of Artificial Intelligence (IJ-AI)*, 2023.
[14] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, https://arxiv.org/abs/1409.1556.
[15] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *Proceedings of the 2017 international conference on engineering and technology (ICET)*, pp. 1–6, IEEE, Antalya, Turkey, August 2017.
[16] S. Sharma, *Artificial Neural Network (ANN) in Machine Learning*, Data science central, Chicago, IL, USA, 2017.
[17] H. Ismanto, A. Azhari, S. Suharto, and L. Arsyad, "Classification of the mainstay economic region using decision tree method," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 3, pp. 1037–1044, 2018.
[18] N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, "Decision tree analysis on j48 algorithm for data mining," *Proceedings of international journal of advanced research in computer science software engineering*, vol. 3, no. 6, pp. 1114–1119, 2013.
[19] N. anaN and V. thri, "Performance and classification evaluation of J48 algorithm and Kendall's based J48 algorithm (KNJ48)," *International Journal of Computer Trends and Technology*, vol. 59, no. 2, pp. 73–80, 2018.
[20] A. Raj, "Perfect recipe for classification using logistic regression," 2020, https://towardsdatascience.com/the-perfect-recipe-for-classification-using-logistic-regression-f8648e267592.
[21] A. Okutse, "A Naïve Bayes' probabilistic classifier for modeling the quality of care in a healthcare setting: GRIN verlag," 2020, https://www.hausarbeiten.de/document/535799.
[22] S. Taheri and M. Mammadov, "Learning the naive Bayes classifier with optimization models," *International Journal of Applied Mathematics and Computer Science*, vol. 23, no. 4, pp. 787–795, 2013.
[23] Y. Elyusufi and Z. Elyusufi, "Social networks fake profiles detection using machine learning algorithms," in *Proceedings of the 3rd International Conference on Smart City Applications*, pp. 30–40, Springer, Tetouan Morocco, October 2020.
[24] N. Donges, "Random forest classifier: a complete guide to how it works in machine learning," 2022, https://builtin.com/authors/niklas-donges.
[25] A. A. Salih and M. B. Abdulrazaq, "Combining best features selection using three classifiers in intrusion detection system," in *Proceedings of the 2019 International Conference on Advanced Science and Engineering (ICOASE)*, pp. 94–99, IEEE, Zakho, Duhok, Iraq, April 2019.
[26] T. Le, D. Tran, W. Ma, T. Pham, P. Duong, and M. Nguyen, "Robust support vector machine," in *Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN)*, pp. 4137–4144, IEEE, Beijing, China, July 2014.
[27] N. Ali, D. Neagu, and P. Trundle, "Evaluation of k-nearest neighbour classifier performance for heterogeneous data sets," *SN Applied Sciences*, vol. 1, no. 12, pp. 1559–1615, 2019.