Optimal Machine Learning Classifier Identification for Facial Recognition

B Pavan Vishnu Sai
Department of Information Technology
Manipal University Jaipur
Jaipur,India
pavanvishnusai@gmail.com

Pallikonda Raj Sudarshan
Department of Mechatronics
Manipal University Jaipur
Jaipur,India
rajsudarshanp@yahoo.com

C Gunal
Department of Computers & Communications
Manipal University Jaipur
Jaipur,India
cgunal2002@gmail.com

Rohit Kumar Gupta
Department of Information Technology
Manipal University Jaipur
Jaipur,India
r.k.7878@gmail.com

***Abstract*— Facial recognition is vital in today's technological world. There is a need to develop facial recognition systems which have high accuracy and at the same time high input to output response. To overcome the problem of low accuracy of facial recognition system we have attempted to combine different algorithm's and develop a high accuracy facial recognition system. On combining different algorithms, we have observed that the Random Forest classifier achieved 96% accuracy in 2 secs. On the other hand, the Linear Discriminant Analysis classifier achieved 97% in 0.34 secs. Here we will compare the 6 classifiers to choose an optimal classifier.**

***Keywords*— Facial Recognition, Support Vector Machine, Random Forest Classifier, Linear Discriminant Analysis, K Nearest Neighbors Classifier, Decision tree, Gaussian Naive Bayes**

I. Introduction

In our daily life facial recognition is a top-notch priority in machine learning and image processing this facial recognition system uses biometrics and other related details to induce information security and law enforcement, thereby increasing the system security and dependability over traditional access control techniques like pins, id cards etc... On the other hand, as machine learning has advanced in recent years, new algorithms and computer resources may be developed to cut run time, increase identification accuracy, and examine additional application situations.

Face recognition technology is widely used, and a variety of commercial face recognition systems may be used [1]. previously we used to develop facial recognition systems such as their main focus was to concentrate on facial characteristics or traits like eye pattern, nose shape, mouth and forehead outlines. this was a huge drawback as the size of the data base grew larger the identification accuracy would decline. This increase in data base size also led to increase in the hardware used [2] .

The other drawback is that this procedure requires perfect start while creating the face model or else there will be security issues. [3] In this newly proposed facial recognition system in a machine learning model we segregate databases into parts, and training and testing of this data is analyzed and recorded. This technique helps us to quantify the accuracy of identification in a facial recognition system. It is also used to analyze the running speed at which the facial recognition system responds using distinct categorization algorithms. [4] [5].

A. ORL Database

The database comprises photos of 40 people, with each participant providing 10 distinct images of themselves as shown in figure1. In lighting conditions such as day and night, wearing (Face mask, sunglasses, hats etc.), and face emotions like (open eyes, laughing, smiling, Captured in different angles). [6]These attributes were taken into consideration. Additionally, all photos can endure a little tilt of up to 20 degrees. Each image is 92x112 pixels in size.

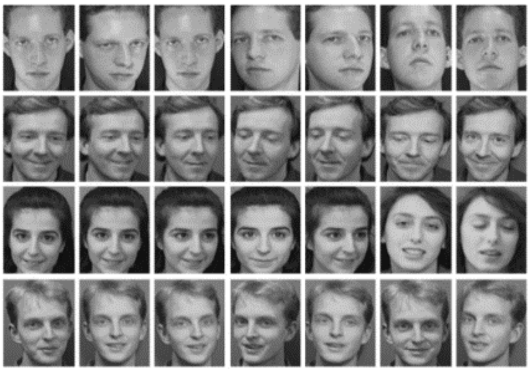

Fig. 1. 40 X 10 ORL database.

II. Related Work

We have started with a dataset from the ORL database and then it will be listed and convert to NumPy array and assign as CLF to categorized using several machine learning approaches such as Random Forest Classifier, Support Vector Machine, Linear Discriminant Analysis, and KNearest Neighbors Classifier, Decision Tree, Gaussian Naïve Byes. The comparison between accuracy and running speed as follows.

A. Random Forest Classifier

In the random forest classifier, we collect data from the distinct decision trees in various subsets and we will average that value thereby, improving the dataset's estimated accuracy. This utilizes the reference of each tree rather than relying on a single decision tree. [7]
RF builds a completely random decision tree from scratch. The algorithm frequently generates great predictions for each iteration. The basic idea behind RF is to determine the average noise value, and complex inputs are thereby calculated into simpler space, which is the goal of RF and

decision trees. [8]shows that the collection of RF, decision trees were trained at random. As a result, in comparison, available data reduces overfitting. As a result, RF technique is a bagging classification tree extension.

It's a parallel learning system. It has a high level of accuracy and a short training period. The following are the advantages of RF [9]:

- Straightforward Learning.
- Local Representation.
- Classification with Occlusion
- Parallelization.
- Fast Training Time.

It combines random selection methods to improve performance, particularly when there are many duplicate features to discriminate [10].

B. Support Vector Machine

The Support Vector Machine (SVM) is a widely used classifier for face recognition because it produces consistent results. They achieved this by creating a hyper plane with the greatest feasible Euclidean distance for the closest qualified examples. Vapnik is the inventor of the support vector machine and its creator (SVM). Because of the qualities and trustworthiness of the empirical results, the SVM is well-known among academics. The SVM method is an excellent classification algorithm [7] .

It's a supervised learning algorithm that's primarily used to divide data into categories. SVM is trained using a set of label data. [11] To divide or classify two classes, SVM draws a decision boundary, which is a hyperplane between them. SVM is also utilized in picture classification and object detection. First, we'll go over the fundamentals of SVMs for binary classification. Then we'll talk about how to apply this strategy to general multi-class classification issues [12].

SVM is a powerful machine learning method that can detect latent consistency in a variety of data sets. Its major function is to improve the generalization ability of learning machines. The support vector refers to those specimens from the training data that are closest to the classified hyperplane, and SVM specifies the vectors of the data feature in the feature space.

C. Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is also known as Discriminant Function Analysis (DFA) or Normal Discriminant Analysis (NDA), is a commonly used dimensionality reduction technique to solve the problems of more than two classes in the machine learning [13] .We use an LDA to convert actual data to free space in an optimal linear transformation. LDA is a linear transformation that can be used to segregate classes in a reduced dimension space. We can easily transform a 2-D and 3-D graph into a 1-dimensional plane using Linear Discriminant Analysis.

Consider the following scenario: we have two classes in a 2-D plane with an X-Y axis that we need to classify quickly. [14] LDA allows us to construct a straight line that entirely separates the two classes of data points, as demonstrated in the previous example. LDA creates a new axis from an X-Y axis by separating them with a straight line and projecting data onto it.

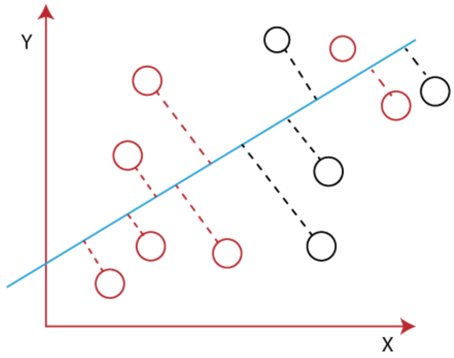

Fig. 2. LDA 2D.

D. K Nearest Neighbor

K-Nearby Neighborhood is a very simple supervised learning-based machine learning technique. The K-NN method thinks that the new case / data and current cases are comparable and assigns it to a category that is extremely similar to the existing categories. The K-NN method organizes all available data and categorizes new data points depending on how similar they are to previous data.

This means that the most recent data may be quickly sorted into the proper category using the K-NN method. [15] Although the K-NN technique is widely employed for taxonomic tasks, it is also utilized for regression. The K-NN algorithm is a non-parametric algorithm that does not make any data predictions.

It's also known as a lazy learning algorithm since it doesn't learn from the training set right away; instead, it saves the dataset and uses it to classify. The KNN approach preserves the dataset throughout the training phase and classifies it into a category that is comparable to the new data when fresh data is received.

E. Decision Tree

Decision Tree is of two types of leaf node and decision node. Decision tree is easy to read classification model because of its accuracy in all kinds of model's application context and with energy-based applications. Decision tree comes under the supervised learning algorithm in machine learning. In decision tree the data is divided into many small-small parts until it reaches its accuracy. such as nodes, leaf nodes and edges. Decision tree is a data mining-based algorithm which is presented for image preprocessing like noise removal.

In decision tree each leaf represents labels with instance. Instances are navigated from the roots to the leaf nodes. [16]Until the classification of all the data in the training data this process will be repeated continuously for the subtree from the new node till all are classified.

To Build a Decision tree we will use CART algorithm (Classification and Regression Tree Algorithm). we can easily understand the whole process of the Decision tree Because it is a tree like structure.

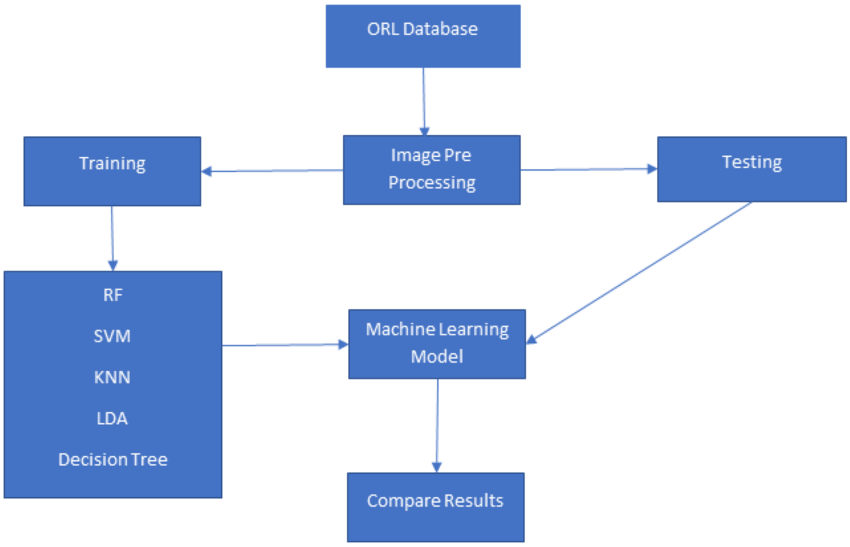

Fig. 3. Methodology of Face Recognition.

F. *Gaussian Naïve Bayes*

Gaussian naive bayes is a logical machine learning algorithm. Naive bayes classification is a collection of classifier algorithm depends on the bayes theorem. naive bayes is a family of algorithm where every algorithm shares a common idea. In Gaussian naive bayes values are continuously assigned to each feature with respect to gaussian distribution. [17]
Gaussian naive bayes is an extension of naive bayes. Gaussian naive bayes is the simple algorithm to calculate the mean and SD of the training data set.

III. Methodology

There are two parts to the face recognition system: database classification and machine learning classification. This phenomenon is shown by the Figure 3 given below.

The ORL faces databases are 40x10, with 40 people and 10 photographs for each. The size of the training and testing data has an impact on detection accuracy First, we will import all the libraries cv2, os, NumPy, matplotlib.pyplot , train_test_split, accuracy_score.

Before going to the next process, we must put our images in order. Then we will reshape the dimensions to 80x 80. Then we will assign dataset to the list. "X_train = list()"
This list is converted to numpy array "X_train1 = np.array(X_train)"

Now the shape will be in 3 dimensional: (287,80,80). Then we will reshape it to 2 dimensional: (287,80*80) = (287, 6400)

Again, in testing we will assign to list and converted to numpy array, here we will make the data reshape (123,80,80). which is in 3D we will reshape to 2D (123,6400).

Performance should be better with more training data and less testing data. In practice, however, the size of the testing data will always be greater in size than that of a training data.

IV. Implementation

A. *Random Forest Classifier*

Now we will import RF from sklearn.ensemble ,here we will assign Randomforestclassifier to clf "(clf = RandomForestClassifier())". We have noted the start time then we have assigned the CLF and then further considered the end time.

For speed "print(f"Runtime of the program is {end - start}") "= 2.21secs

For accuracy "clf.score(X_test1, y_test)" = 0.96

B. *Support Vector Machine*

Now we will import SVM from sklearn ,here we will assign SVM to clf1 "clf1 = svm.SVC() ". We have noted the start time then we have assigned the CLF and then further considered the end time. For speed "print(f"Runtime of the program is {end - start}") "= 0.35secs
For accuracy "clf1.score(X_test1, y_test)" = 0.95

C. *Linear Discriminant Analysis*

Now we will import LinearDiscriminantAnalysis from sklearn.discriminant_analysis ,here we will assign LinearDiscriminantAnalysis to clf2 "clf2 = LinearDiscriminantAnalysis()". We have noted the start time then we have assigned the CLF and then further considered the end time.
For speed "print(f"Runtime of the program is {end - start}") "= 0.34 secs
For accuracy "clf2.score(X_test1, y_test)" = 0.97

D. *K Nearest Neighbor*

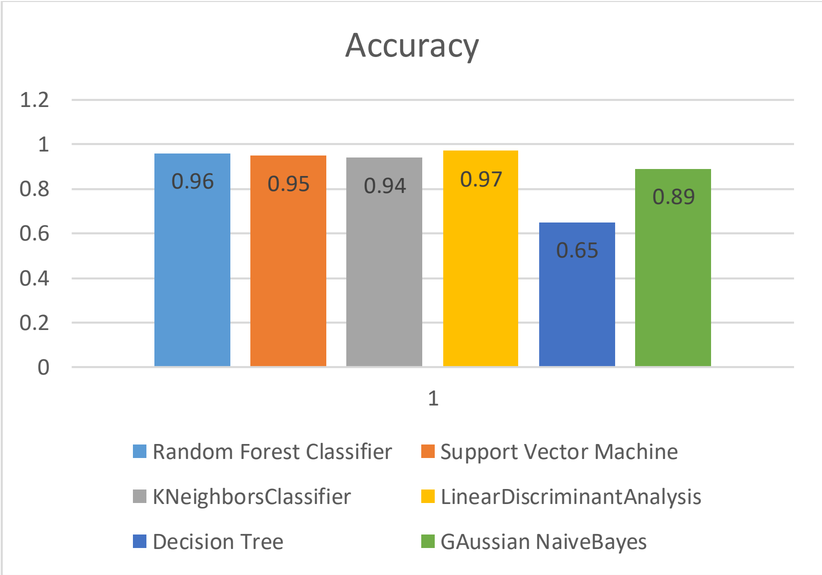

Fig. 4. Accuracy.

KNeighborsClassifier is being imported from sklearn.neighbors, and assigned KNeighbors Classifier to neigh
"neigh = KNeighborsClassifier(n_neighbors=3)".
We have noted the start time then we have assigned the CLF and then further considered the end time.
For speed "print(f"Runtime of the program is {end - start}") "= 0.00099 secs
For accuracy "neigh.score(X_test1, y_test)" = 0.94

E. Decision Tree

Now we will import Decision Tree from sklearn ,here we will assign Decision Tree to clf3 "clf3 = tree.DecisionTreeClassifier()".We have noted the start time then we have assigned the CLF and then further considered the end time.
For speed "print(f"Runtime of the program is {end - start}") "= 3.26 secs
For accuracy "clf3.score(X_test1, y_test)" = 0.65

F. Gaussian Naïve Bayes

Now we will import GAussian NaiveBayes from sklearn.naive_bayes ,here we will assign GAussian NaiveBayes to gnb "gnb = GaussianNB()". We have noted the start time then we have assigned the GNB and then further considered the end time.
For speed "print(f"Runtime of the program is {end - start}") "= 0.33 secs
For accuracy "gnb.score(X_test1, y_test)" = 0.89

V. Experimental Results

TABLE I. Speed & Accuracy Comparision Table

Classifiers	Speed	Accuracy
Random Forest Classifier	2.21	0.96
Support Vector Machine	0.35	0.95
KNeighborsClassifier	0.00099	0.94
LinearDiscriminantAnalysis	0.34	0.97
Decision Tree	3.26	0.65
GAussian NaiveBayes	0.33	0.89

From the above table we can see that Decision tree speed is 3.26 which takes more time compared to every classifier also the accuracy is 0.65 which is least accuracy as we can see in the table so we can ignore and Gaussian Naive Bayes speed is 0.33 which takes least time, but the accuracy is 0.89 which is less. If we compare RF from the table, we can see speed is 2.21 which takes a large amount of time, and the accuracy is 0.96 which is good. SVM speed is 0.35 and the accuracy is 0.95 which is good. KNN takes the least time and accuracy is also good. If we compare LDA speed is 0.34 and the accuracy is 0.97 which is very less time and very good accuracy.

VI. Conclusion

This research compares two facial recognition methods. Using the ORL database and six different types of machine learning classifiers, we were able to achieve reliable accuracy in facial recognition at high running speed. LDA is a top-notch reliable system where there is high accuracy and at decently low running speed. In this LDA classifier we get desirable accuracy in facial recognition system at significantly low running speed. On the other hand, we have Random Forest Classifier which also has a high accuracy but at a cost of high running speed which is not feasible. Also 3rd place we can take SVM and KNN. Other machine learning classifications could be used in future studies, and some of them could be combined to create a more complicated system. But these classifiers should operate at low running speed and sustain high recognition accuracy for larger datasets.

REFERENCES

[1] R. C. W. Zhao, "Face Recognition: A Literature Survey," in *ACM Computing Surveys , Vol.35, No.4*, 2003.

[2] M. A. P. Turk, "Eigenfaces for recognition," *Journal of cognitive neuroscience,* vol. 3.1, pp. 71- 86, 1991.

[3] D. N. B. B. M. Parmar, "Face recognition methods & applications.," *International Journal of Computer Technology & Applications,* vol. 4(1), pp. 84-86, 2013.

[4] A. G. N. K. &. H. T. Amit Verma, "Face Recognition: A Review and Analysis," in *Springer*, Singapore, 2022.

[5] S. B. D. B. S. V. Jash Dalvi, "A Survey on Face Recognition Systems," in *arxiv*, 2022.

[6] G. K. M. P. Harihara Santosh Dadi, "Improved Face Recognition Rate Using HOG Features and SVM," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE,* vol. 11, pp. 34-44, 2016.

[7] E. K. a. A. Subasi, "Performance of Random Forest and SVM," in *The International Arab Journal of Information Technology VOL. 13,*, 2016.

[8] B. L, "Random Forests," *Machine Learning,* vol. 45(1), pp. 5-32, 2001.

[9] V. BELLE, Detection and Recognition of Human Faces using Random Forests for a Mobile Robot, Aachen: Rheinisch-Westalishce Technische Hochschule Aachen, 2008., 2008.

[10] K. R. Archer K, "Empirical Characterization of Random Forest Variable Importance Measures," *Computational Statistics and Data Analysis,* vol. 52(4), pp. 2249-2260, 2008.

[11] W. M. B. H. a. Z. L. A. Adouani, "Comparison of Haarlike, HOG and LBP approaches for face detection in video sequences," in *16th International Multi-Conference on Systems, Signals & Devices (SSD)*, Istanbul, Turkey, 2019.

[12] D. E. a. H. S. Z. B. Lahaw, "Robust Face Recognition Approaches Using PCA, ICA, LDA Based on DWT, and SVM Algorithms," in *Conference on Telecommunications and Signal Processing (TSP)*, Athens, 2018.

[13] C. H. H. P. Park, "A comparison of generalized linear discriminant analysis algorithms," *Pattern Recognition,* vol. 41(3), pp. 1083-1097, 2008.

[14] A. E. W. a. C. B. A. M. Chihaoui, "A Survey of 2D Face Recognition," in *Computers, vol.5, no. 4*, 2016.

[15] M. H. M. A. F. H. A.-M. Mustafa Zuhaer Nayef Al-Dabagh, "Face Recognition System Based on Kernel Discriminant Analysis,K-Nearest Neighbor and Support Vector Machine," *International Journal of Research and Engineering,* vol. Vol.5 No.3, pp. 335-338, 2018.

[16] S. J. B. Tajinder Kumar, "Face Recognition with Decision Tree using SVM and SURF," *international journal of control theory and application,* vol. 10, pp. 173-180, 2017.

[17] S. Sharma, M. Bhatt and P. Sharma, "Face Recognition System Using Machine Learning Algorithm," in *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, Coimbatore, India, 2020.

[18] L. S. M. Kirby, "Application of the Karhunen-Loeve Procedure for the characterization of human faces," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12(1), pp. 103-108, 1990.

[19] F. A. a. A. P. C. Maaoui, "Wavelet transform based facial feature," in *14th International Multi-Conference on Systems, Signals* , 2017.