Face Recognition using Traditional Classifiers

Shreekar Mane[1] Tavishi Srivastava[1] Debadatta Sahoo[1] Mayank Agarwal[1]
Dhruv Mishra[1] Ekta Saini[1]

[1]Indian Institute of Technology, Jodhpur
{b23cs1069, b23cs1101, b23cm1013, b23cm1052, b23cs1090, b23cs1018}@iitj.ac.in

Abstract

Face identification is a crucial task in computer vision, widely used in security systems and identity verification. This study evaluates the effectiveness of various machine learning techniques, including Decision Trees, K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Random Forest, Artificial Neural Networks (ANN), and Clustering, in face identification using different types of processed data. We extract features from images utilizing Convolutional Neural Networks (CNN) with ResNet, Histogram of Oriented Gradients (HoG), and Local Binary Patterns (LBP) to enhance identification accuracy. Additionally, Linear Discriminant Analysis (LDA) is applied to reduce dimensionality and improve feature separability. Our research systematically analyzes the accuracy and computational efficiency of these techniques on processed datasets. The findings indicate that while Decision Trees remain simple, they perform competitively with well-processed facial features. Similarly, SVM, KNN, Random Forest, ANN, and Clustering provide valuable insights into face identification, showcasing the advantages and trade-offs of different approaches. These results suggest that both traditional and neural network-based machine learning techniques can serve as efficient and scalable solutions for face identification applications.

Keywords: Face Identification, Machine Learning, Decision Trees, KNN, Clustering, SVM, Random Forest, ANN

Contents

1 Introduction

Face identification is a critical task in the field of computer vision. The objective of face identification is to determine the identity of a person based on their facial features, often from a given dataset of labeled images.

In this project, we are performing Face Identification using the Labeled Faces in the Wild (LFW) dataset — a well-known benchmark containing 13,233 images of 5,749 different faces collected from the web, with variations in pose, lighting, expression, and background. To address this problem, we implemented and compared a variety of machine learning algorithms, including Decision Tree, K-Nearest Neighbors (KNN), Clustering (unsupervised learning), Random Forest, Artificial Neural Networks (ANN), and Naive Bayes. Each algorithm was evaluated based on its accuracy, performance, and suitability for the task of face identification.

Feature extraction techniques like CNN and LBP were used to convert image data into a suitable format for model training and evaluation.

Our major findings indicate that while complex models like ANN provide higher accuracy, simpler models such as KNN and Random Forest can also yield competitive results with lower computational requirements. Decision Tree and Clustering methods also gave a considerable accuracy, given the skewness of the data, however, Naive Bayes algorithm did not give an accuracy which could make our model preferable.

The rest of the report is organized as follows:

- Section 2 provides a detailed overview of the approaches tried to solve the problem.
- Section 3 discusses the Experiments and Results obtained (Visualisations).
- Section 4 provides a summary of the overall project.
- Lastly, the contributions of each group member are mentioned.

2 Approaches Tried

This section describes various approaches that were tried in the implementation of model for Face Identification.

2.1 Decision Tree [4]

2.1.1 With CNN features

The Decision Tree model was applied using features extracted from images utilizing Convolutional Neural Networks (CNN). The model was trained with a maximum depth of 6. The results were as follows:

- **Test Accuracy:** 0.75
- **Train Accuracy:** 0.99

The high training accuracy indicates that the model fits the training data well, while the test accuracy suggests a reasonable generalization to unseen data.

2.1.2 With LBP Features

Another approach involved using features extracted through Local Binary Patterns (LBP). The Decision Tree model was trained with a maximum depth of 3. The results were:

- **Test Accuracy:** 0.47
- **Train Accuracy:** 0.65

The lower accuracy in both training and testing phases suggests that LBP features may not be as effective for this task compared to CNN features.

These experiments highlight the impact of feature extraction techniques on the performance of Decision Tree models in face recognition tasks.

2.2 K-Nearest Neighbours [1]

2.2.1 With CNN features

The KNN model was applied using features extracted from images utilizing Convolutional Neural Networks (CNN).
KNN on CNN features only (keeping k=5):

- **Test Accuracy:** 57.14%
- **Train Accuracy:** 80.00%

We applied Dimensionality Reduction methods such as PCA and LDA as well to the data for making better predictions. The results were as follows:
On applying PCA (keeping k=5):

- **Test Accuracy:** 57.14%
- **Train Accuracy:** 79.76%

On applying LDA (keeping k=5):

- **Test Accuracy:** 75.23%
- **Train Accuracy:** 96.90%

The high Train accuracy and low Test accuracy indicates that the model fits the training data very well (overfitting), however, it is not a well-generalized model to give good predictions on Test Data.

2.2.2 With LBP Features

Another approach involved using features extracted through Local Binary Patterns (LBP).
KNN on LBP features only:

- **Test Accuracy:** 30.476%
- **Train Accuracy:** 63.09%

Again, Dimensionality Reduction methods such as PCA and LDA were applied. The results were as follows:
On applying PCA (keeping k=5):

- **Test Accuracy:** 32.38%
- **Train Accuracy:** 63.80%

On applying LDA (keeping k=5):

- **Test Accuracy:** 32.38%
- **Train Accuracy:** 95.71%

A much lower Test accuracy and high Train accuracy in both PCA and LDA methods suggests that LBP features may not be as effective for this task compared to CNN features.

2.2.3 With Combined CNN and LBP Features

Another approach involved combining the CNN and LBP features using concatenation.The results were:
KNN on Combined Features only (best k = 7):

- **Test Accuracy:** 60.00%
- **Train Accuracy:** 78.09%

We applied Dimensionality Reduction on Combined Features as well.

On applying PCA (best k = 7):

- **Test Accuracy:** 60.00%
- **Train Accuracy:** 78.33%

On applying LDA (best k = 3):

- **Test Accuracy:** 79.04%
- **Train Accuracy:** 98.09%

From the above results, we can see that applying KNN on Combined Features gives better results (as compared to using only CNN or only LBP features).

The best test accuracy is obtained in the last case, wherein we have applied LDA on Combined CNN and LBP features.

These experiments highlight the impact of feature extraction techniques on the performance of KNN models in Face Identification tasks.

2.3 Clustering

[2] Trained and tested the model on CNN extracted features and LBP extracted features. Result shown is for k=5

2.3.1 Clustering on CNN features

- **Test Accuracy:** 77.00%
- **Train Accuracy:** 100.00%

2.3.2 Clustering on LBP features

- **Test Accuracy:** 36.40%
- **Train Accuracy:** 73.68%

2.4 Random Forest

The Random Forest ensemble method was implemented using two distinct feature extraction techniques to compare performance characteristics.

2.4.1 With LBP Features

- **Test Accuracy:** 0.50
- **Train Accuracy:** 1.00

2.4.2 on applying LDA

- **Test Accuracy:** 0.49
- **Train Accuracy:** 1.00

there is no significant increase in performance after applying LDA, this is because LBP features are shallow and texture based.[3].

2.4.3 With CNN Features

- **Test Accuracy:** 0.57
- **Train Accuracy:** 0.99

2.4.4 on applying LDA

- **Test Accuracy:** 0.802
- **Train Accuracy:** 1.00

CNN features give same accuracy as to lbp features initially but after applying lda the accuracy improves by more than 20%

2.5 Naive Bayes

2.5.1 With CNN features

Naive Bayes was implemented from scratch using CNN-extracted features. The dataset was filtered to include labels with at least 80 samples, and feature scaling was applied. The model assumes Gaussian distribution of features and calculates class-wise likelihoods accordingly.

- **Test Accuracy:** 0.39
- **Train Accuracy:** 0.57

The model showed moderate performance, indicating limited generalization ability, likely due to the independence assumptions of Naive Bayes and the high dimensionality of CNN features.

2.5.2 With LBP Features

Using LBP features, the same Naive Bayes implementation was applied. After filtering and scaling, the model produced:

- **Test Accuracy:** 0.28
- **Train Accuracy:** 0.39

The lower accuracy suggests that LBP features are less effective for face recognition tasks under the Naive Bayes framework.

These results show that CNN features offer better performance than LBP, although Naive Bayes may not be the most suitable model for complex image-based classification tasks.

2.6 Artificial Neural Network

The ANN method was implemented on both the CNN features data and the LBP features data.

2.6.1 With CNN features on all classes

Applied the ANN method on the original CNN features data with all the classes and features.

- **Test Accuracy:** 45.98
- **Train Accuracy:** 100

2.6.2 With CNN features on 5 classes

The classes which have a minimum of 80 samples in the data were used.

- **Test Accuracy:** 80.43
- **Train Accuracy:** 100

2.6.3 With CNN features and applying LDA

- **Test Accuracy:** 79.33
- **Train Accuracy:** 100

The test accuracies of this data with and without applying LDA because after applying LDA, the model becomes incapable to classify 5 classes with 4 features, less and skewed data.

2.6.4 With LBP Features

- **Test Accuracy:** 57.75
- **Train Accuracy:** 99.90

2.6.5 With LBP Features and applying LDA

- **Test Accuracy:** 52.60
- **Train Accuracy:** 84.06

This shows that the LBP data is not suited for this task.

3 Experiments and Results

This section provides details about the dataset, experimental setup, and result comparisons. Additionally, graphs illustrating the relationship between tree depth and accuracies are included.

3.1 Dataset and Experimental Setup

3.1.1 Dataset

The dataset used for this study consists of facial images processed using various feature extraction techniques, including CNN, HoG, and LBP. For this experiment, CNN and LBP extracted data were utilized. A filter was applied to select individuals with more than 80 images, resulting in a subset of 5 different persons, each with over 80 images. Linear Discriminant Analysis (LDA) and principle component analysis (PCA) was then applied to reduce dimensionality and improve feature separability.

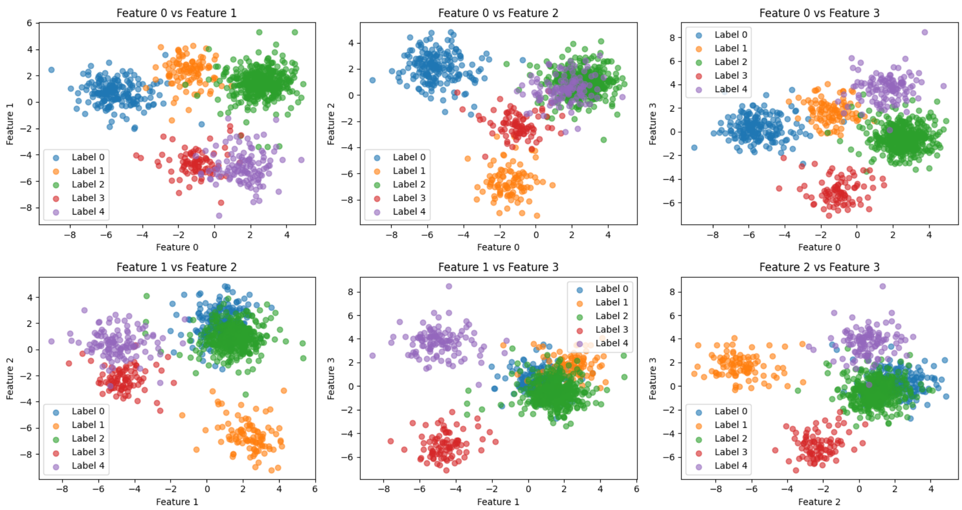

Figure 1: Projection of train data along two feature dimensions after LDA

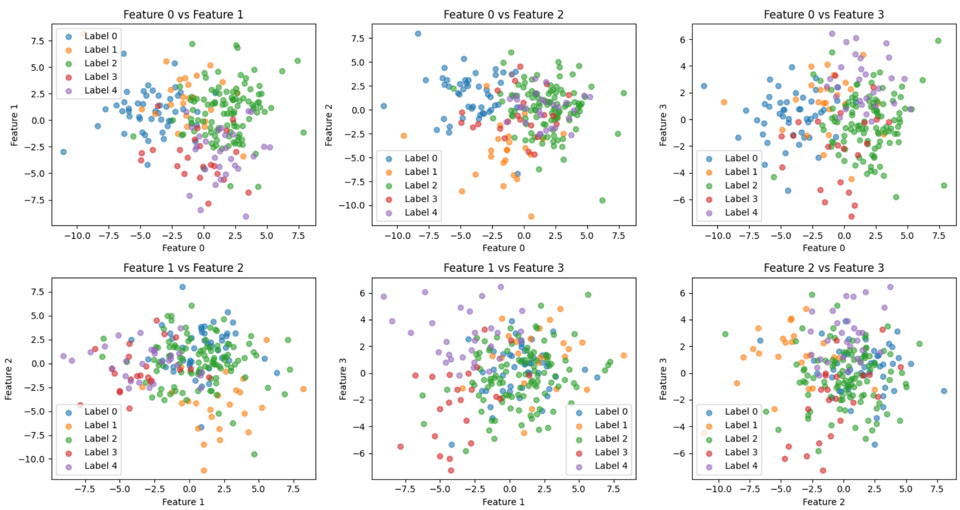

Figure 2: Projection of train data along two feature dimensions after LDA

3.1.2 Experimental Setup

The experiments were conducted using different ML models with varying its hyper-parameters to assess their impact on accuracy. Feature extraction was performed using CNN and LBP. Linear Discriminant Analysis (LDA) was applied for dimensionality reduction.

3.2 Results and Analysis

3.2.1 Decision Tree with CNN Features

- **Test Accuracy:** 0.75
- **Train Accuracy:** 0.99

The high training accuracy indicates that the model fits the training data well, while the test accuracy suggests a reasonable generalization to unseen data.

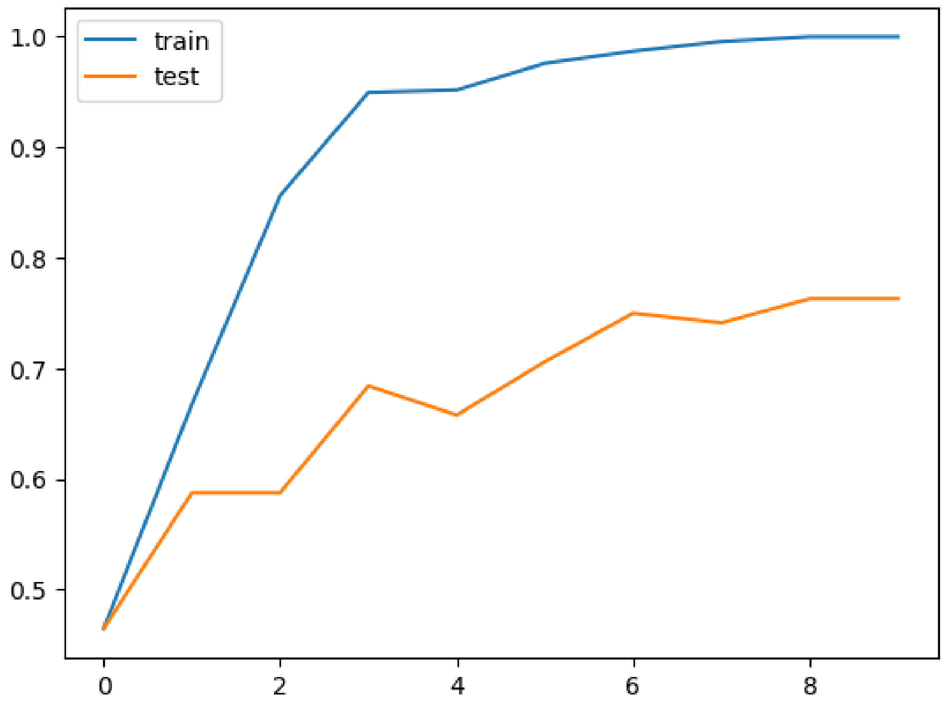

Figure 3: Depth of tree vs. Accuracy for CNN Features with LDA

These graphs demonstrate how the choice of tree depth affects the model's performance, providing insights into the trade-offs between complexity and accuracy.

3.2.2 Decision Tree with LBP Features

- **Test Accuracy:** 0.47
- **Train Accuracy:** 0.65

The lower accuracy in both training and testing phases suggests that LBP features may not be as effective for this task compared to CNN features.

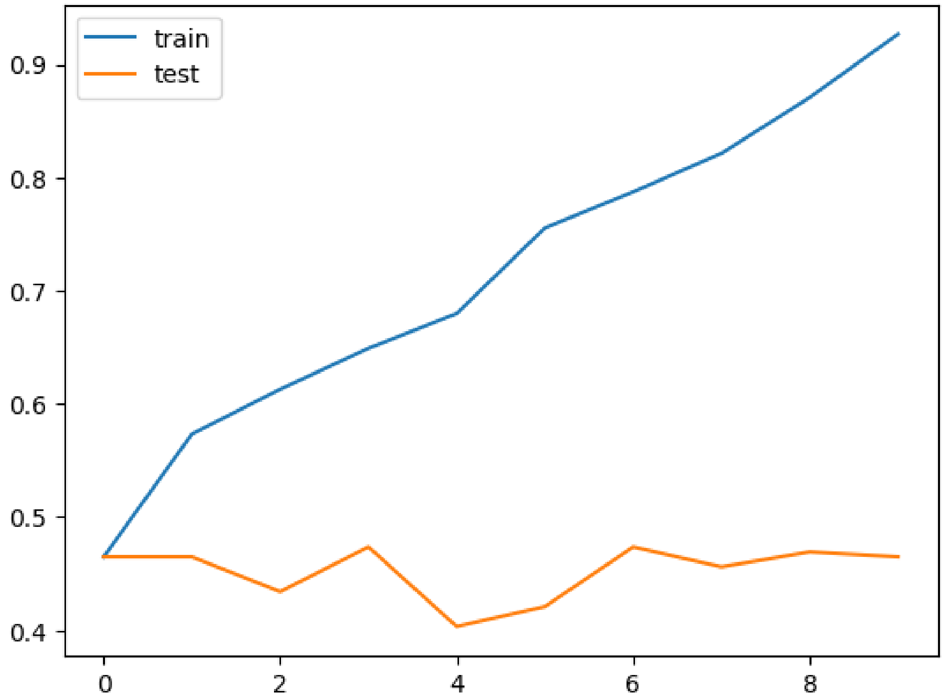

Figure 4: Depth vs. Accuracy for LBP Features

These graphs (figure 4 and figure 3 demonstrate how the choice of tree depth affects the model's performance, providing insights into the trade-offs between complexity and accuracy.

3.2.3 K-Nearest Neighbours

Since, the KNN model worked best when applied to combined CNN and LBP features, we have plotted the graphs of 'k' vs accuracy and Confusion Matrices for the Combined features case. The visualisations are as follows :

1. **On Combined features (without Dimensionality Reduction) :**

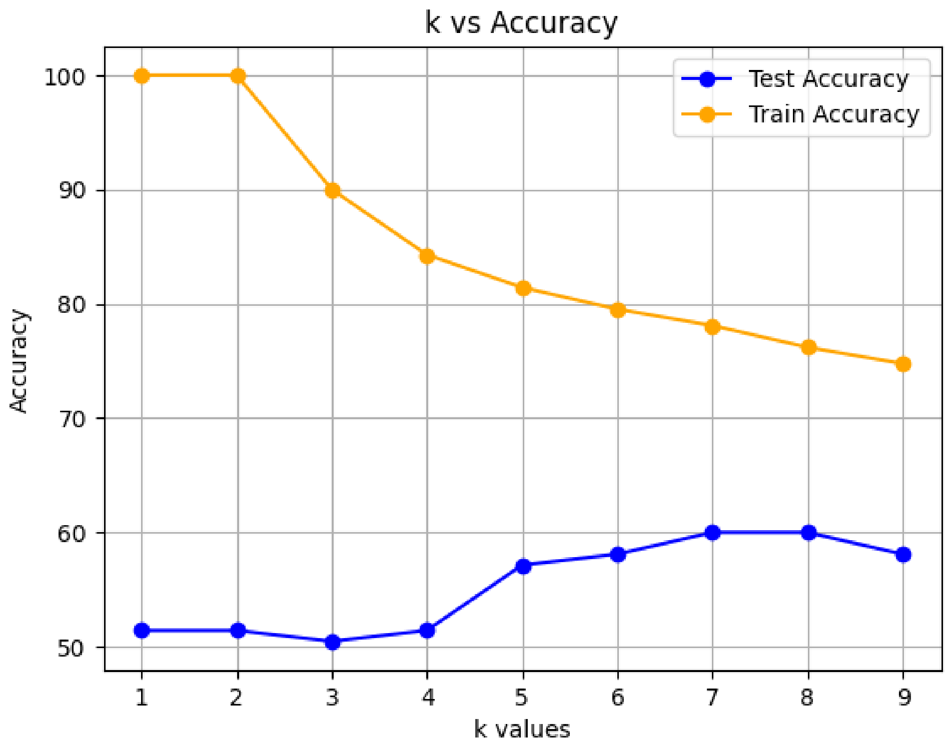

Figure 5: k vs. Accuracy for Combined features (without Dimensionality Reduction)

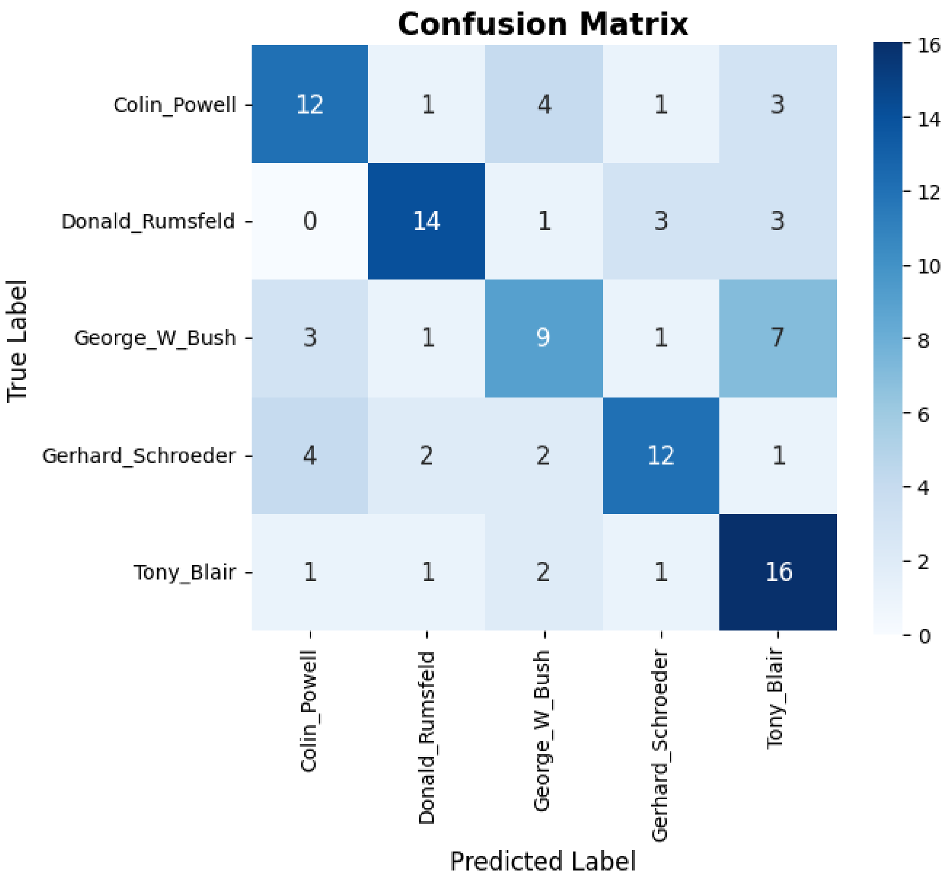

Figure 6: Confusion Matrix (no Dimensionality Reduction)

2. **On Combined features (applying PCA) :**

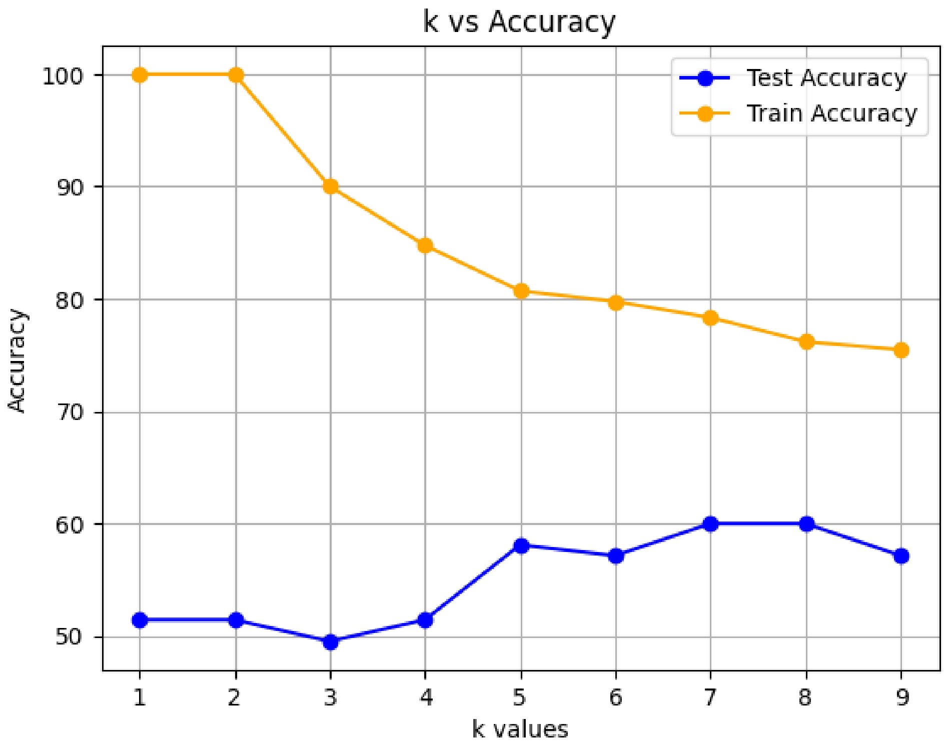

Figure 7: k vs. Accuracy for Combined features (after applying PCA)

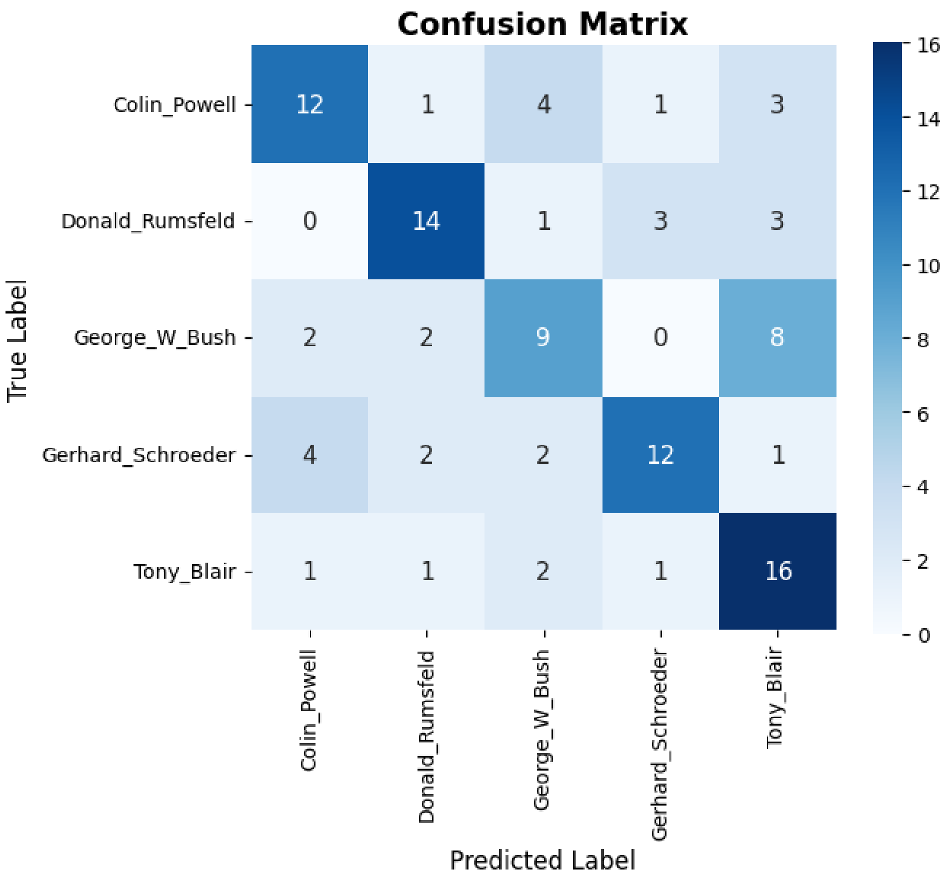

Figure 8: Confusion Matrix (PCA applied)

3. **On Combined features (applying LDA) :**

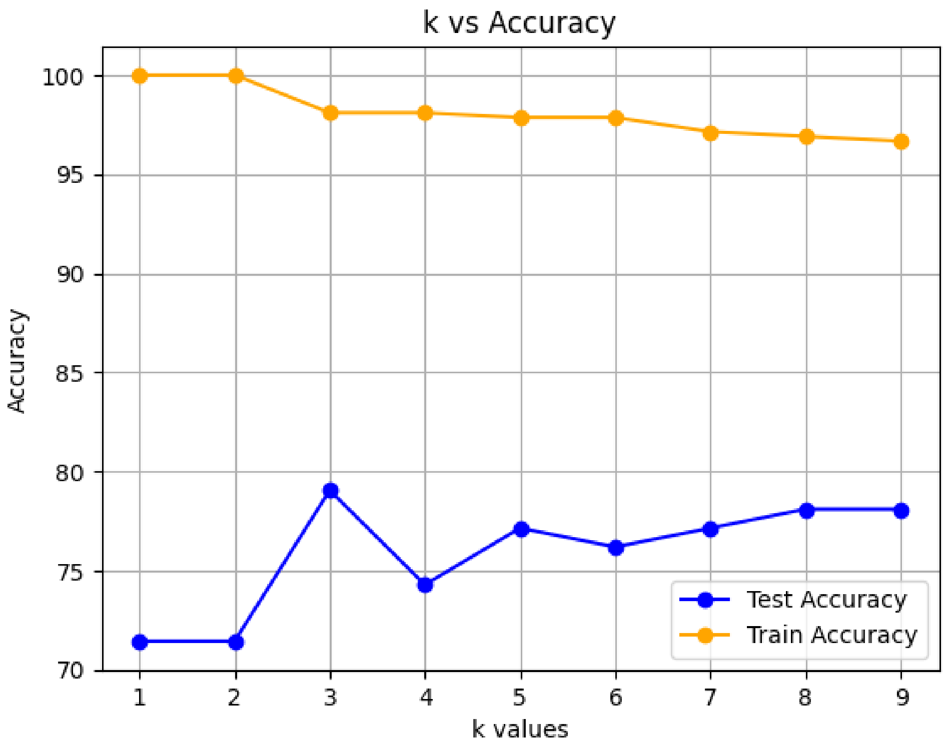

Figure 9: k vs. Accuracy for Combined features (after applying LDA)

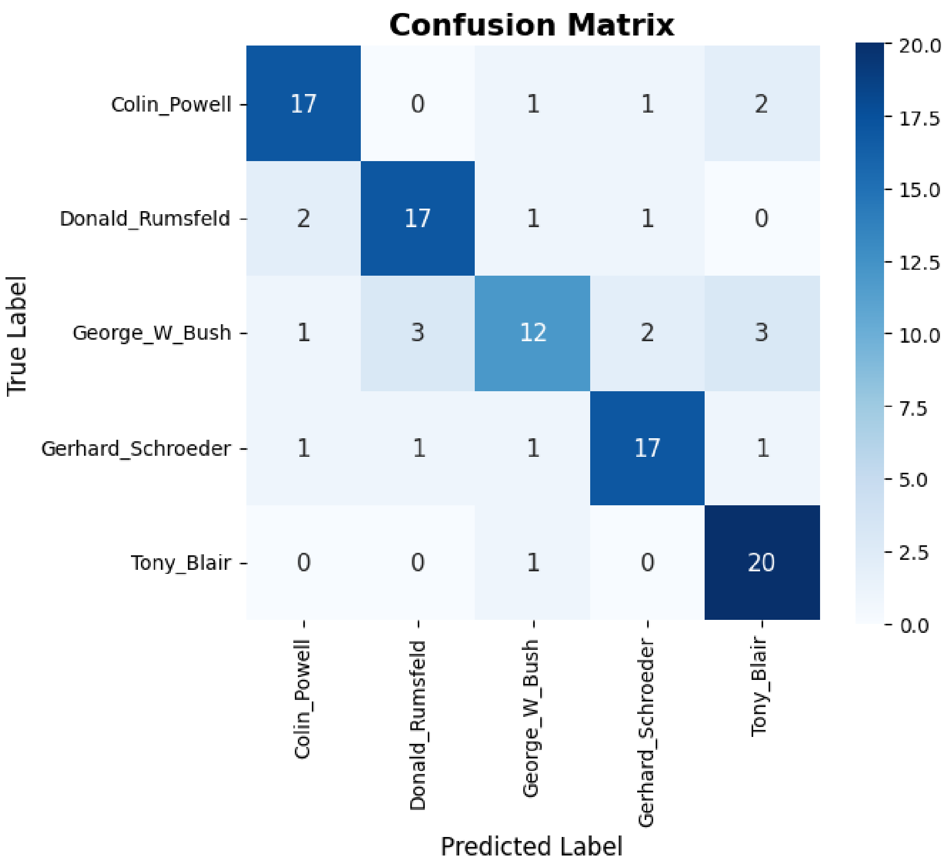

Figure 10: Confusion Matrix (LDA applied)

The above graphs and Confusion Matrices help us compare the different methods used for Feature Extraction and Dimensionality Reduction, thereby letting us choose the best method through which a high Test accuracy is obtained, and the number of True Positives are increased.

From our observation, it is clear that the case when LDA is applied on the combined CNN and LBP features, and KNN classification is performed, we get the highest number of True Positives.

This method had a Test accuracy = 79.04% and a Train accuracy = 98.09%

Now, applying LDA on Combined features and reducing features to 2, in order to plot the actual labels and the predicted labels on a 2-D graph and compare the results.

- **Test Accuracy:** 63.81%
- **Train Accuracy:** 83.57%

Actual Labels :

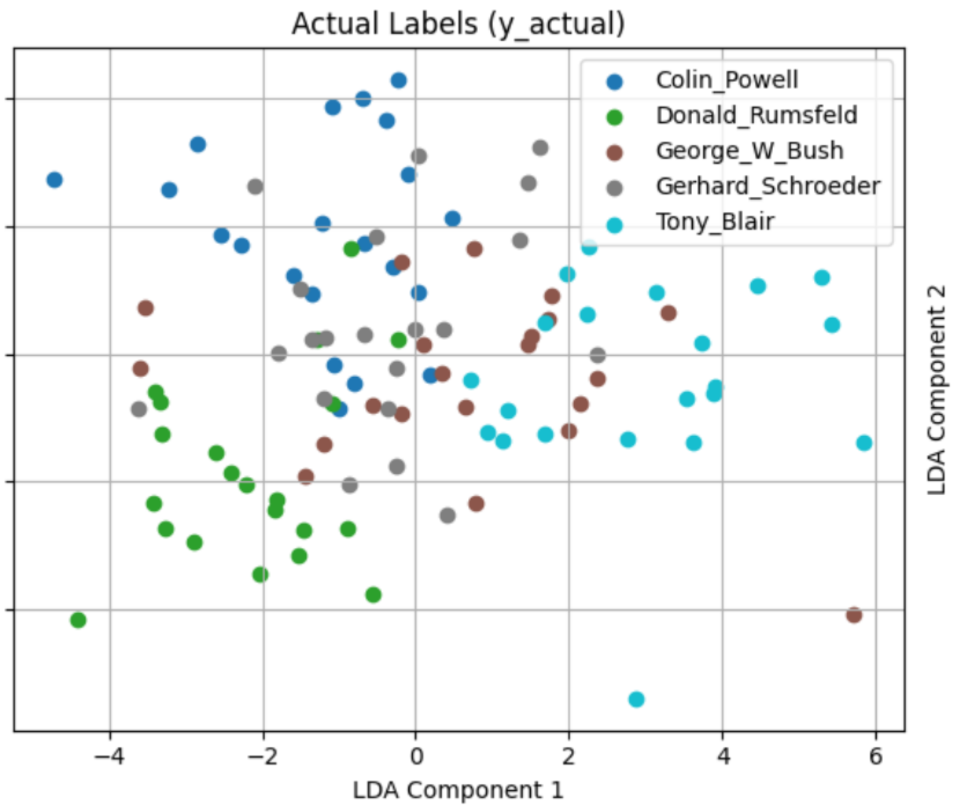

Figure 11: Actual labels

Predicted Labels :

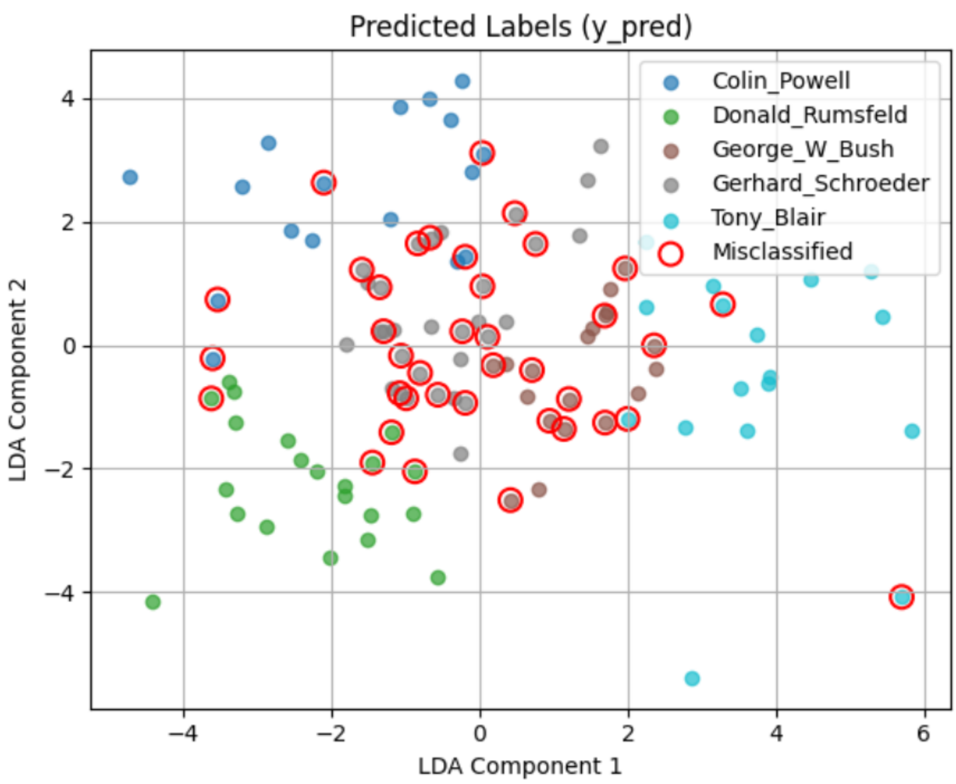

Figure 12: Predicted labels

The above plots help us identify the misclassifications in the predicted labels (marked in red).

3.2.4 Clustering

1. **On CNN Extracted features -** High performance on both training and test sets.
   - Well-formed clusters in LDA space with meaningful separability.
   - Indicates effective generalization and resistance to overfitting.
   - Best suited for unsupervised clustering applications.

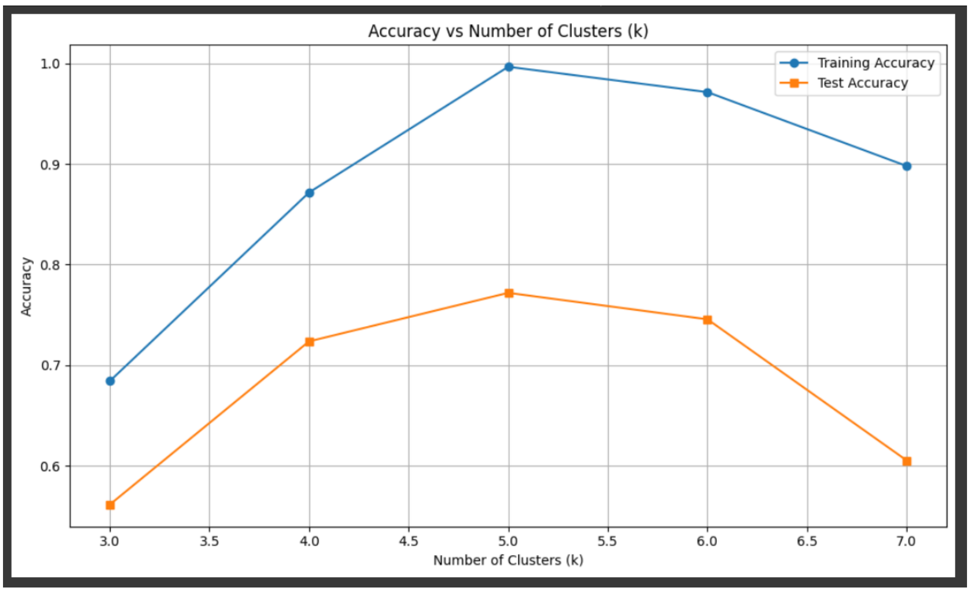

Figure 13: K vs. Accuracy for CNN Features

2. **On LBP Extracted features** - High training accuracy but poor test accuracy, indicating overfitting.
- Poor generalization due to reliance on shallow, texture-based features.
- LDA transformation may distort feature space in a way that reduces robustness.
- Not ideal for applications requiring strong feature generality.

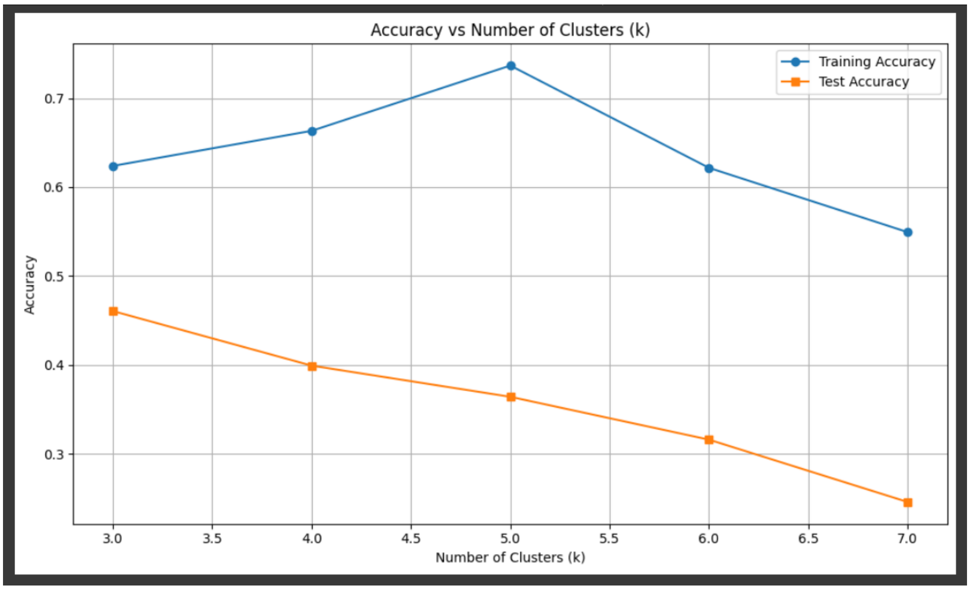

Figure 14: K vs. Accuracy for LBP Features

CNN features generalize well with consistent clustering and minimal misclassifications, showing strong robustness to facial variations. In contrast, LBP features overfit the training data, perform poorly on test samples, and produce fragmented, less reliable clusters.

3.2.5 Naive Bayes with CNN Features

- **Test Accuracy:** 0.39
- **Train Accuracy:** 0.57

The moderate training and relatively low test accuracy suggest that Naive Bayes struggles to capture the complexity of CNN-extracted features. This is likely due to the model's strong assumption of feature independence, which may not hold in high-dimensional CNN data.

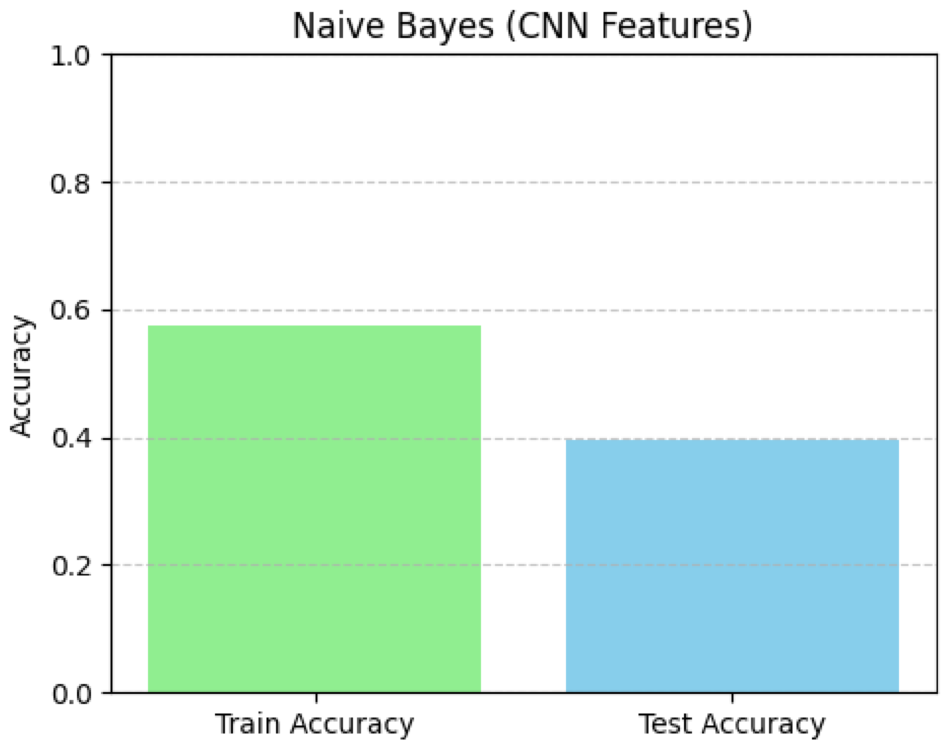

Figure 15: Accuracy Comparison for CNN Features using Naive Bayes

3.2.6 Naive Bayes with LBP Features

- **Test Accuracy:** 0.29
- **Train Accuracy:** 0.39

LBP features led to even lower accuracy in both training and testing, indicating that they are less effective in this context. The model's assumptions and the simplicity of LBP descriptors contribute to this performance drop.

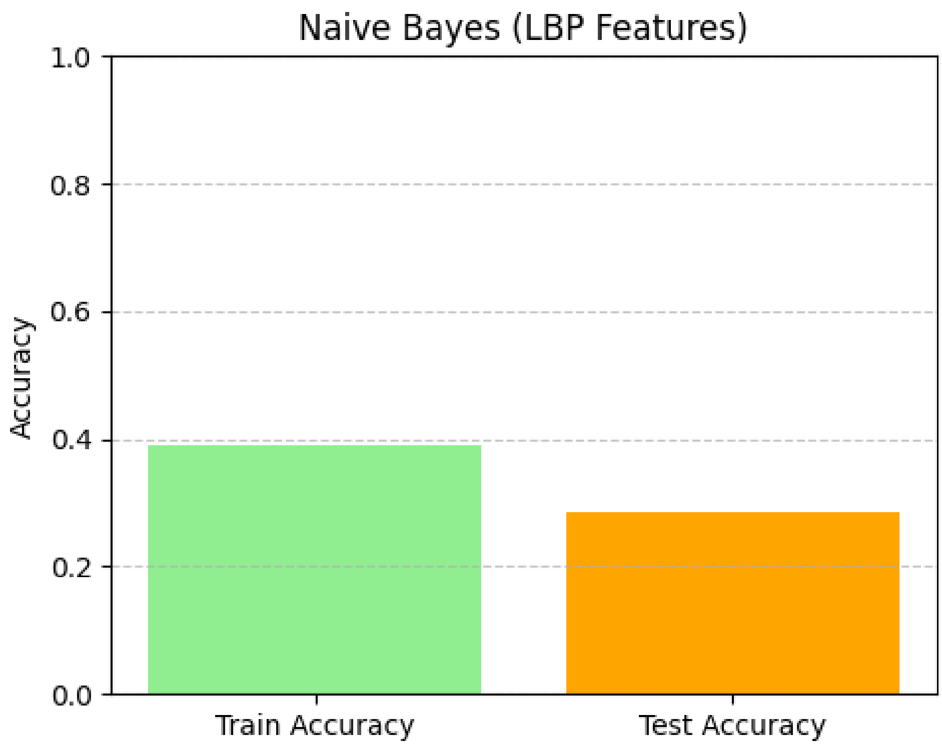

Figure 16: Accuracy Comparison for LBP Features using Naive Bayes

These results show that Naive Bayes is less effective for face recognition compared to Decision Trees, particularly when using simpler features like LBP. The choice of feature extraction method significantly impacts performance, and CNN features consistently outperform LBP in both models.

3.2.7 Random forest with CNN features

- **Test Accuracy:** 0.80
- **Train Accuracy:** 1.00

Improvement over Decision tree by 5%.high training accuracy means it fits well to training data and reasonable test accuracy indicates that the model has not overfitted. these results are shown when number of trees are 100 and depth of each tree is 10.

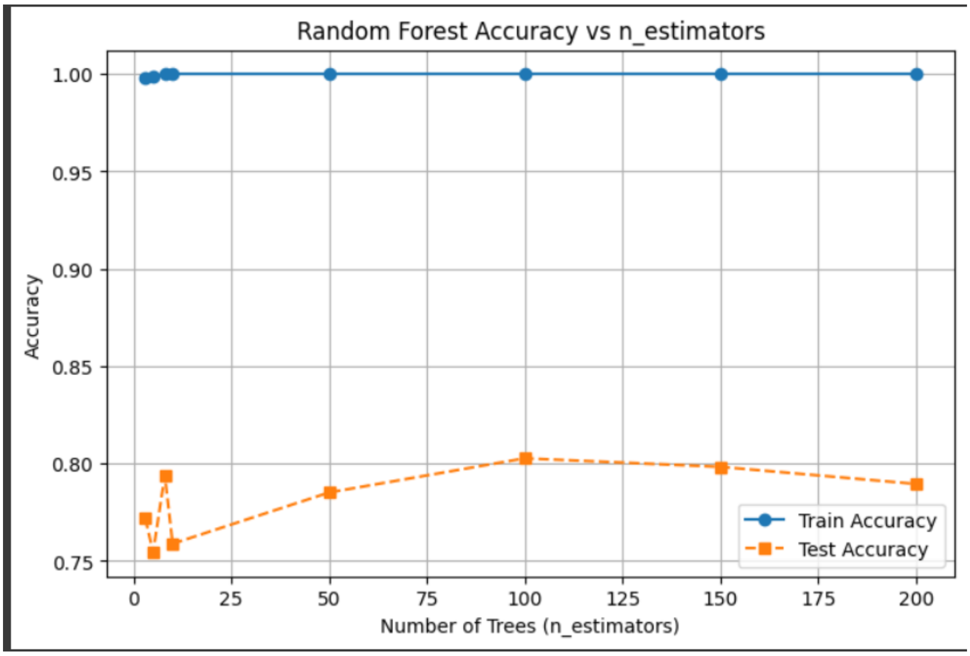

Figure 17: Accuracy Comparison for CNN Features using random forest

3.2.8 Random forest with LBP features

- **Test Accuracy:** 0.5044
- **Train Accuracy:** 1.00

despite train accuracy being good test accuracy is below average indicating that LBP features do not provide enough information for the model and are shallow features.However Random forest improves over decision tree by 3

In both LBP and CNN features n=100 shows best test and train accuracy

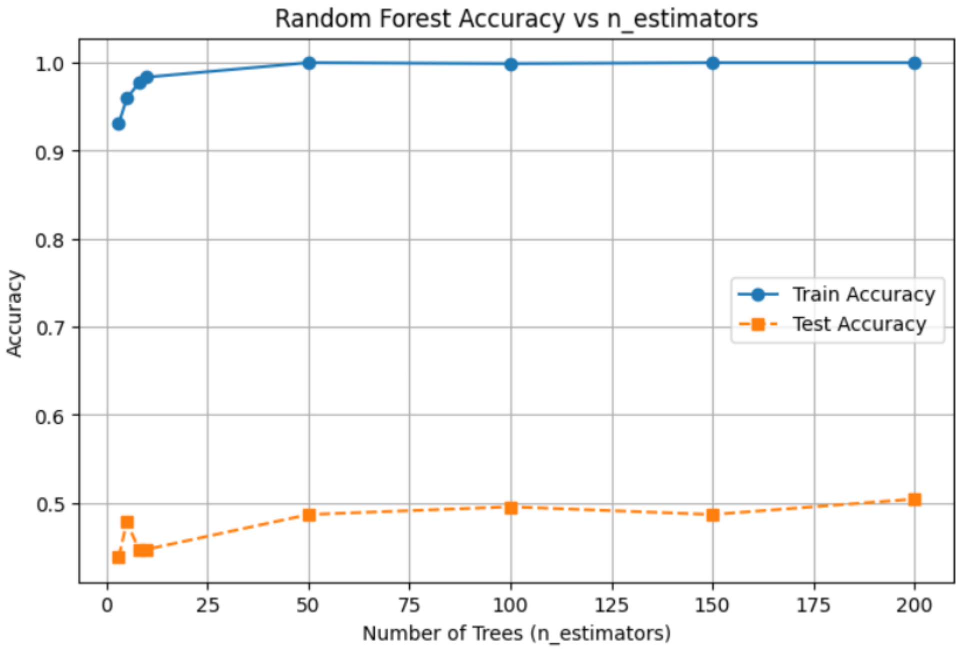

Figure 18: Accuracy Comparison for LBP Features using random forest

3.2.9 ANN with CNN features

- **Test Accuracy:** 80.43
- **Train Accuracy:** 100

ANN was applied on CNN with and without LDA. Furthermore without LDA it was applied without filtering the data that is on all the classes and with filtering that is the minimum sample should be 80 and it performed better in the latter one. The model overfits in all the 3 versions due to less samples and don't give a good accuracy due to high skewness.

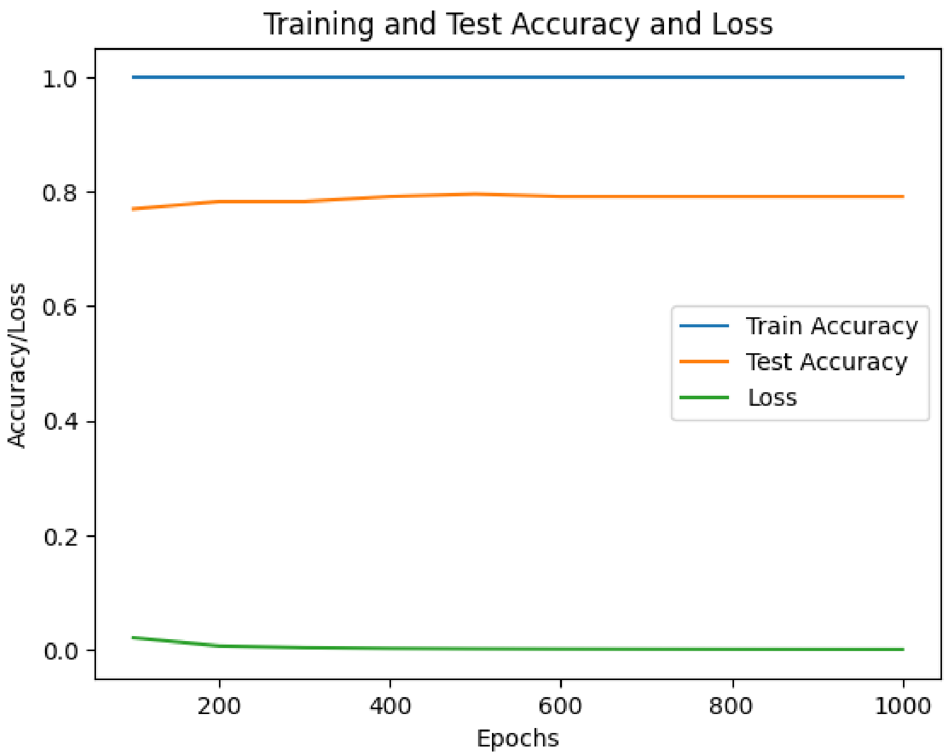

Figure 19: Accuracy/loss Comparison for CNN Features using ANN

3.2.10 ANN with LBP features

- **Test Accuracy:** 52.60
- **Train Accuracy:** 84.61

Despite train accuracy being good, test accuracy is not satisfying which is mainly due to the insuitability of LBP features for CNN tasks while ANN performed better than rest all models.

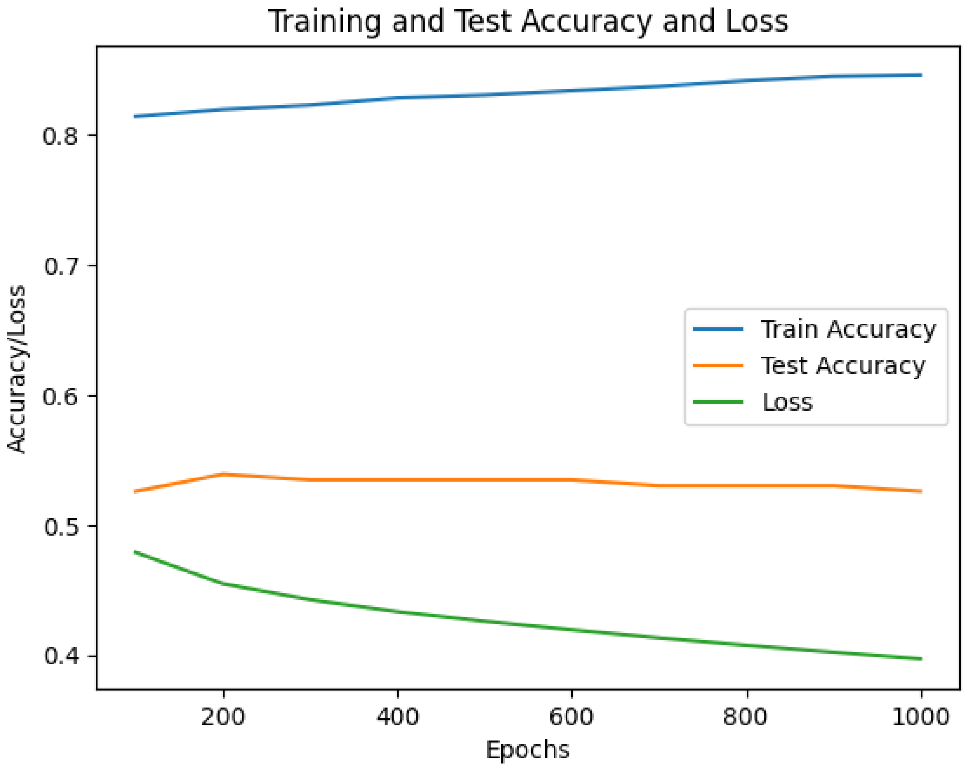

Figure 20: Accuracy/loss Comparison for LBP Features using ANN

4 Summary

In this project, we explored the problem of Face Identification using the Labeled Faces in the Wild (LFW) dataset by implementing and comparing several traditional machine learning algorithms. The primary goal was to evaluate and analyze the effectiveness of these algorithms on image data processed with different feature extraction methods.

We applied feature extraction techniques such as Convolutional Neural Networks (CNN) and Local Binary Patterns (LBP), and further used dimensionality reduction techniques like Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) to improve model performance. The classifiers evaluated include Decision Tree, K-Nearest Neighbors (KNN), Clustering (K-Means), Random Forest, Naive Bayes, and Artificial Neural Networks (ANN).

From our experiments, it was observed that CNN features consistently outperformed LBP features across all classifiers. Among all models, the best test accuracy of 80.43% and 80% was achieved when ANN and Random Forest respectively were applied to filtered CNN features. The KNN model also gave a high test accuracy of 79.04% when applied on combined CNN and LBP features followed by LDA. However, Naive Bayes showed the lowest performance, particularly with LBP features, due to its simplistic assumptions and the high dimensionality of the data.

These results indicate that traditional classifiers can perform effectively when paired with powerful feature extraction and dimensionality reduction techniques. The comparative analysis provides insights into the strengths and limitations of each model, laying the groundwork for future improvements in face identification systems.

References

[1] Idelette Laure Kambi Beli and Chunsheng Guo. Enhancing face identification using local binary patterns and k-nearest neighbors. *Journal of Imaging*, 3(3):37, 2017.

[2] R. Kumar, R. Rajput, and R. Saini. Real time 2-d face detection using color ratios and k-mean clustering. In *Proceedings of the 8th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pages 255–262. ACM, 2006.

[3] HUDA. H. Mady and Shadi M. S. Hilles. Face recognition and detection using random forest and combination of lbp and hog features. *2018 International Conference on Smart Computing and Electronic Enterprise (ICSCEE)*, pages 1–7, 2018.

[4] Daniel Maturana, Domingo Mery, and Álvaro Soto. Face recognition with decision tree-based local binary patterns. In *Asian Conference on Computer Vision*, pages 618–629. Springer, 2010.

A Contribution of Each Member

1. Shreekar Mane: Implemented data preprocessing for CNN features, Implemented Decision Tree model and prepared the project-page and prepared web-demo.
2. Tavishi Srivastava: Implemented data preprocessing for LBP features, implemented KNN classification technique, prepared web-demo.
3. Debadatta Sahoo: Implemented Clustering model.
4. Mayank Agarwal: Implemented Artificial Neural Network model.
5. Dhruv Mishra: Implemented Random Forest model.
6. Ekta Saini: Implemented Naive Bayes approach.

B Special Thanks to Google

We used the virtual machine service from Google Cloud to deploy the web demo of our project. Thanks to the Google Cloud Coupon provided through the Google Cloud Teaching Award, we were able to access this resource for free. This allowed us to host our website efficiently without any cost, making deployment simple and effective. We sincerely thank Google for their support in enabling us to use cloud infrastructure at no expense.