Face Recognition Model Training and Analysis

Hai Nam LE

1 Extended Learning

1.1 Objectives of Training a Learning Model for Face Recognition

The goal of training a face recognition model is to accurately identify or verify individuals based on facial features. Key objectives include:

- **Dimensionality Reduction:** Reduce the high-dimensional data to manage computational complexity.
- **Feature Extraction:** Identify informative features that distinguish individuals.
- **Class Discrimination:** Maximize separation between different classes (individuals).
- **Real-time Performance:** Ensure the model operates efficiently for practical applications.

1.2 Role of Linear Discriminant Analysis (LDA) in Face Recognition

LDA is a supervised dimensionality reduction technique used to enhance class separability. It focuses on maximizing the ratio of between-class variance to within-class variance, making it highly effective for classification tasks.

- **Fisherfaces:** Features extracted using LDA, optimized for distinguishing between individuals.
- **Advantages:**
  - Enhances class separation, leading to better recognition performance.
  - Requires fewer components than PCA, reducing computational costs.
- **Limitations:**
  - Requires class labels and assumes linear separability.
  - Less effective with a very high-dimensional dataset without preprocessing.

1.3 Comparison Between LDA and PCA in Face Recognition

Table 1: Comparison Between PCA and LDA in Face Recognition

Aspect	**PCA (Principal Component Analysis)**	**LDA (Linear Discriminant Analysis)**
Objective	Maximize data variance.	Maximize class separability.
Supervision	Unsupervised; no class labels.	Supervised; uses class labels.
Features Produced	Eigenfaces: Captures most variance in data.	Fisherfaces: Maximizes discrimination between classes.
Dimensionality Reduction	Can reduce to any number of principal components.	Reduces to $\min(\text{n\_classes} - 1, \text{n\_features})$.
Focus	Captures global data structure.	Focuses on discriminative features.
Applications	Effective for noise reduction and redundancy removal.	Best for classification tasks.

1.3.1 Visualization: Eigenfaces vs. Fisherfaces

- **Eigenfaces** represent the principal components from PCA. They capture the main variance in the dataset, showing abstract and blurred facial features. Eigenfaces focus on overall variations in lighting, pose, and facial structure without considering class separability. The images are less distinct and appear more ghost-like, emphasizing areas of general variance.

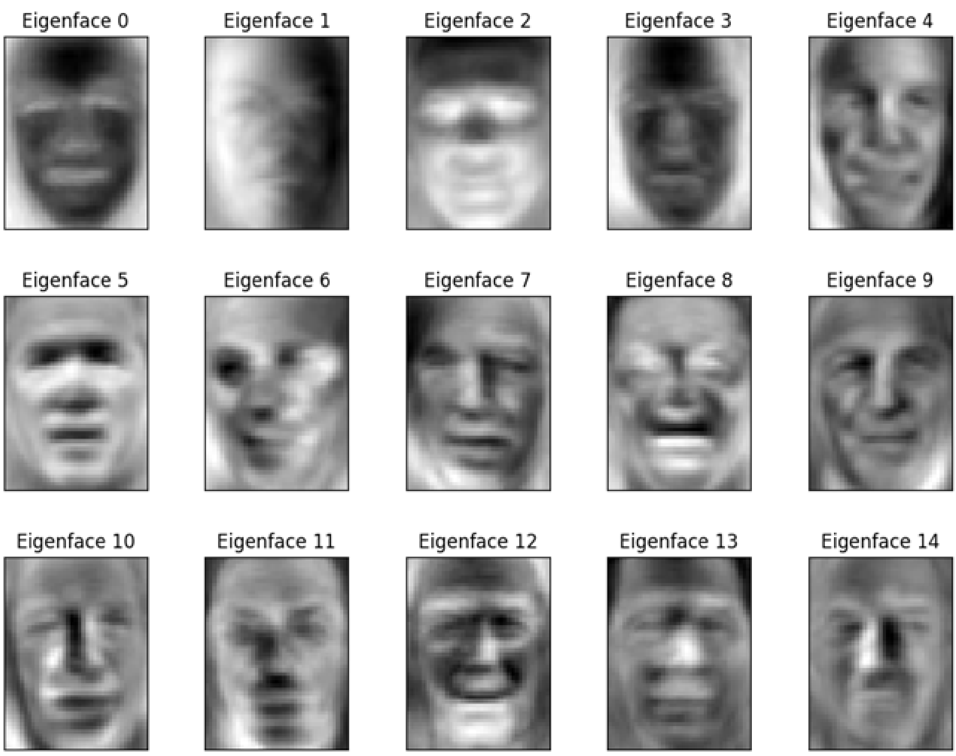

Figure 1: Eigenfaces representing generalized patterns such as lighting effects, face orientation, and general facial features.

- **Fisherfaces** highlight the most discriminative features that separate different classes (faces of different individuals). Fisherfaces show sharper and more distinct facial features like eyes, nose, and mouth, focusing on the variations that best distinguish one person from another. They are more defined than eigenfaces, with clearer contours and contrast in key facial regions.

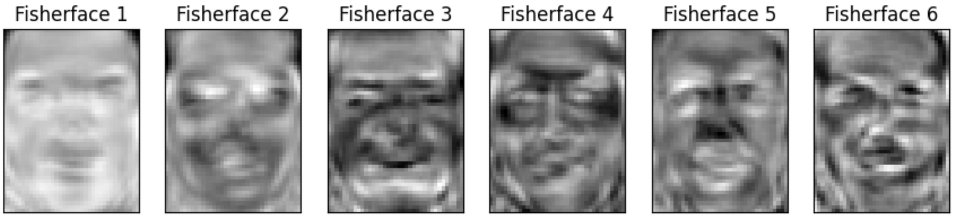

Figure 2: Fisherfaces showing the most discriminative facial features for distinguishing between individuals

1.3.2 Differences in Performance of LDA and PCA

PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) perform differently due to their distinct objectives:

- **PCA** is an unsupervised technique that identifies the directions of maximum variance in data. It projects data onto a new subspace to retain as much variance as possible. This makes PCA effective for visualization, compression, and understanding natural patterns in data. PCA is more suitable when you need to reduce dimensionality without specific regard to class labels. It can handle multicollinearity but does not necessarily focus on separating classes.

- **LDA** is a supervised technique that maximizes class separability by projecting data onto a subspace that minimizes within-class variance while maximizing between-class variance. LDA is designed to extract the most discriminative features for classification tasks, making it more effective than PCA when the goal is to distinguish between classes. However, LDA requires more samples per class to prevent overfitting and is sensitive to class imbalances and outliers.

1.3.3 Practical Impact of Variance on Class Separability

- **Variance in PCA:** PCA focuses on capturing the overall variance in the data, which can include both between-class and within-class variations. If most variance is due to factors that are not relevant to class separability (e.g., lighting or background), PCA may not provide the best features for distinguishing between classes.

- **Variance in LDA:** LDA directly targets variance that improves class separability by considering class labels. It seeks to "stretch" the differences between classes and "squash" variations within the same class, making it more suitable for classification tasks where distinguishing between different groups is crucial.

1.3.4 When to Use PCA or LDA

Use PCA When:

- We have a large dataset with no class labels and want to explore high-dimensional data.
- The goal is to reduce noise and redundancy or perform generic feature extraction before modeling.
- The dataset has a high variance that is informative for exploratory analysis.
- We need to reduce the risk of overfitting by reducing dimensions without considering class labels.

Use LDA When:

- The dataset has class labels, and the goal is to maximize class separation for classification tasks.
- The dataset has more samples per class, and you need to optimize feature extraction for class discrimination.
- We aim to improve model performance by finding projections that enhance class separability.

2 Application

2.1 Significance of Using the LFW Dataset in Face Recognition

The Labeled Faces in the Wild (LFW) dataset is a well-known benchmark for face recognition research. It contains over 13,000 labeled images of faces collected from the web, featuring various individuals in different poses, lighting conditions, and expressions.

- **Real-world Variability:** The LFW dataset includes faces with diverse expressions, orientations, and lighting, making it a good proxy for real-world conditions.
- **Class Imbalance:** The dataset is imbalanced, with some individuals having many images and others having only a few. This challenges models to generalize well.
- **Benchmarking:** It is widely used for benchmarking face recognition algorithms, allowing for performance comparisons across different methods and research papers.

2.2 Preprocessing Techniques Applied

Preprocessing is crucial to prepare the dataset for dimensionality reduction and classification. The following steps were applied:

- **Resizing:** Images were resized to a uniform dimension (resize=0.4) to standardize input data for the model.
- **Grayscale Conversion:** LFW images are already in grayscale, simplifying the preprocessing pipeline.

- **Data Standardization:** Each feature (pixel intensity) was standardized to have zero mean and unit variance. This step ensures that PCA and LDA perform optimally by removing bias due to varying scales of features.

```
from sklearn.datasets import fetch_lfw_people
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.discriminant_analysis import
   LinearDiscriminantAnalysis as LDA
from sklearn.svm import SVC
from sklearn.metrics import classification_report,
   ConfusionMatrixDisplay
from scipy.stats import loguniform
from sklearn.model_selection import RandomizedSearchCV
import time

# Load the LFW dataset
lfw_people = fetch_lfw_people(min_faces_per_person=70, resize
   =0.4)

# Data and target arrays
X = lfw_people.data
y = lfw_people.target
target_names = lfw_people.target_names

# Shape parameters
n_samples, h, w = lfw_people.images.shape
n_features = X.shape[1]
n_classes = target_names.shape[0]

print(f"n_samples: {n_samples}, n_features: {n_features},
   n_classes: {n_classes}")

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y,
   test_size=0.25, random_state=42)

# Standardize the data
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

2.3 Visualizations and Their Significance

- **Distribution of Samples per Class:**

  - **Description:** This bar chart shows the number of images per individual in the Labeled Faces in the Wild (LFW) dataset.

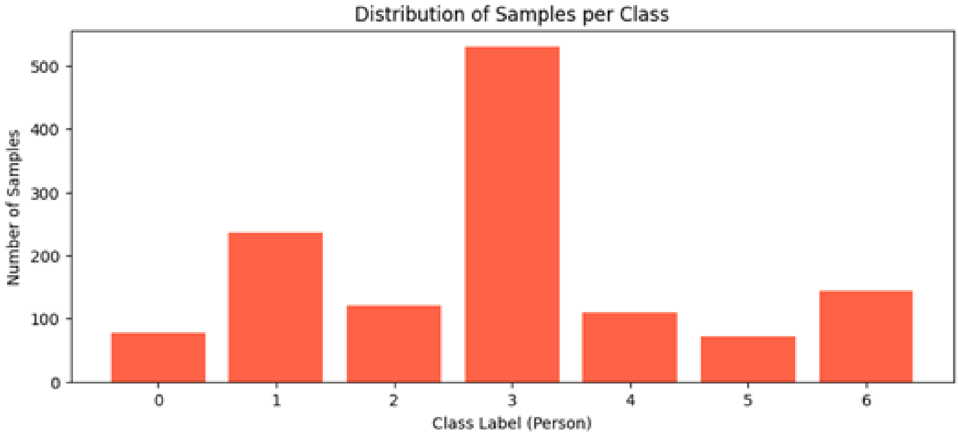

Figure 3: Distribution of samples per class in the LFW dataset.

  - **Significance:** It highlights class imbalance, where some individuals have significantly more images. Understanding this imbalance is crucial for building robust models that do not overfit to well-represented classes.

- **PCA vs. LDA 2D Projections:**
  - **Description:** Scatter plots show the LFW dataset reduced to two dimensions using PCA (left) and LDA (right).

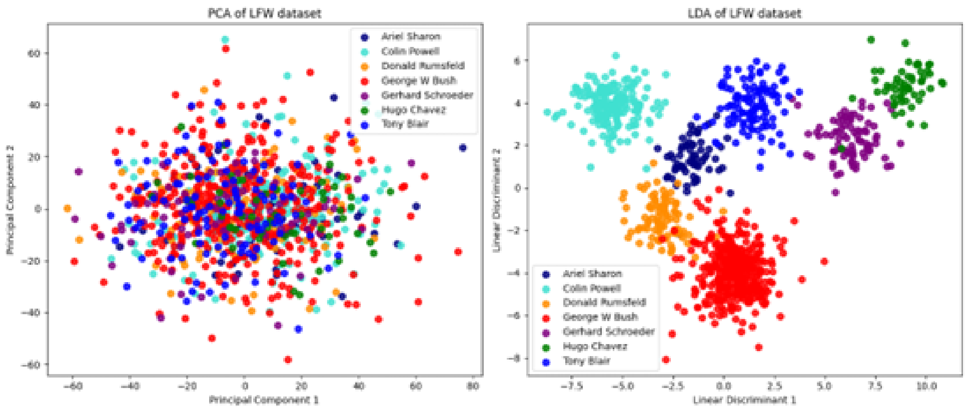

Figure 4: PCA vs. LDA projections of the LFW dataset.

  - **Significance of PCA:** PCA captures the maximum variance in the data but does not focus on class separation, as indicated by overlapping clusters. It's good for reducing dimensionality but less effective for tasks requiring distinct class separation.
  - **Significance of LDA:** LDA maximizes class separability, resulting in distinct clusters for each individual, which is crucial for face recognition tasks. The clear separation of clusters demonstrates LDA's effectiveness in distinguishing between different individuals.

- **Correlation Matrix of Pixel Intensities:**
  - **Description:** Heatmap showing correlations between pixel intensities across images.

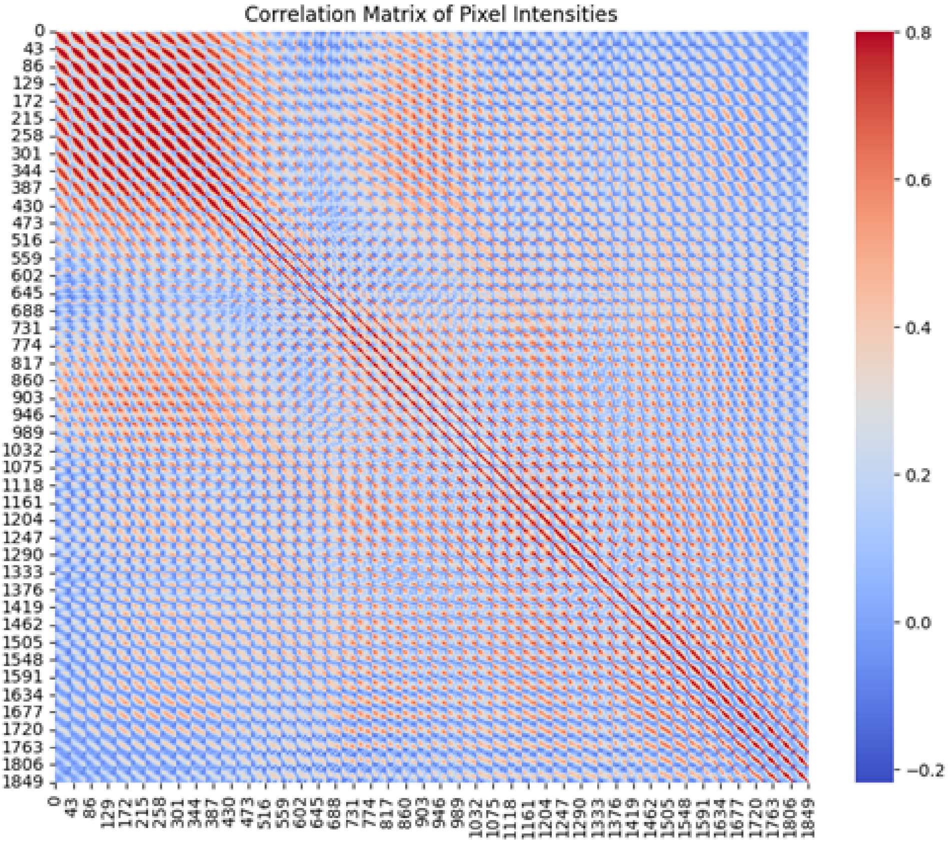

Figure 5: Correlation matrix of pixel intensities in the LFW dataset.

  - **Significance:** Identifies which pixels vary together, suggesting they belong to similar facial regions. This helps in feature selection, focusing on informative regions like eyes and mouth, and removing noise.

2.4 Implementing LDA on the LFW Dataset

Linear Discriminant Analysis (LDA) is implemented to reduce dimensionality while maximizing class separability. The number of components is set to $n_{\text{classes}} - 1$ to ensure optimal separation.

```
from sklearn.discriminant_analysis import
   LinearDiscriminantAnalysis as LDA

# Apply LDA
lda = LDA(n_components=n_classes-1)
X_train_lda = lda.fit_transform(X_train, y_train)
X_test_lda = lda.transform(X_test)
```

Key Procedures:

- **Fitting LDA:** The `fit_transform` method computes the optimal projection matrix that maximizes class separability, transforming the training data into a lower-dimensional space.
- **Transforming Test Data:** The `transform` method projects the test data using the learned projection matrix.

2.5 Classifier Used for Model Building: SVM

For classification, we use a Support Vector Machine (SVM) with a radial basis function (RBF) kernel. SVMs are well-suited for high-dimensional spaces and effective in cases where the number of dimensions exceeds the number of samples.

Why SVM?

- **Robust to Overfitting:** SVMs are effective in high-dimensional spaces and robust to overfitting, especially with a properly chosen kernel.
- **Kernel Trick:** The RBF kernel allows the SVM to model non-linear decision boundaries, which is useful for complex datasets like LFW.
- **Optimization with Grid Search:** We use `RandomizedSearchCV` for hyperparameter tuning (C and gamma), optimizing the classifier's performance.

```
from sklearn.discriminant_analysis import
   LinearDiscriminantAnalysis as LDA

# Apply LDA
lda = LDA(n_components=n_classes-1)
X_train_lda = lda.fit_transform(X_train, y_train)
X_test_lda = lda.transform(X_test)

# Train SVM with LDA
param_grid = {'C': loguniform(1e3, 1e4), 'gamma': loguniform
   (0.0001, 0.001)}
clf_lda = RandomizedSearchCV(SVC(kernel='rbf', class_weight='
   balanced'), param_grid, n_iter=10, cv=5,\
                              random_state=42)
clf_lda.fit(X_train_lda, y_train)

# Predict and evaluate LDA-based model
y_pred_lda = clf_lda.predict(X_test_lda)
print("LDA-based Model Performance:")
print(classification_report(y_test, y_pred_lda, target_names=
   target_names))

ConfusionMatrixDisplay.from_estimator(clf_lda, X_test_lda,
   y_test, display_labels=target_names, xticks_rotation=45)
plt.show()
```

3 Comprehensive Analysis

3.1 Baseline Model (No Dimensionality Reduction)

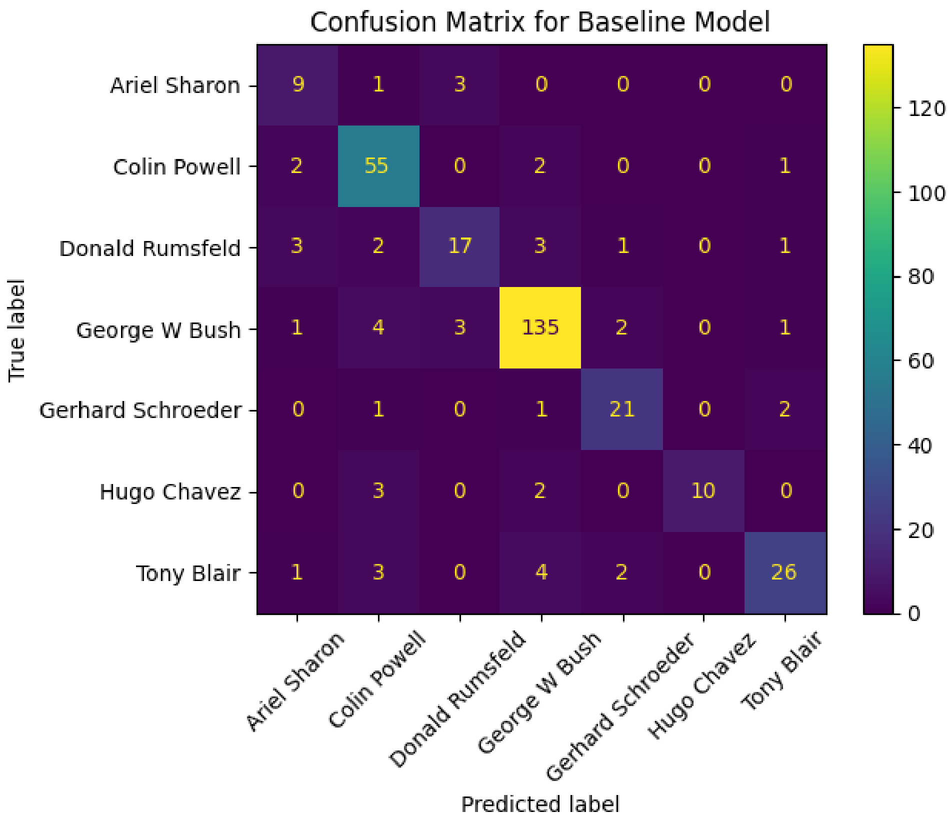

Figure 6: Confusion Matrix for Basesline Model.

Performance Metrics:

- Accuracy: 85%
- Weighted Average Precision: 85%
- Weighted Average Recall: 85%
- Weighted Average F1-Score: 85%

Analysis:

- The baseline model without any dimensionality reduction shows the highest accuracy (85%) among the three approaches. This suggests that the SVM classifier is capable of distinguishing faces effectively when using all available features.
- However, using the full dimensionality (all pixels) can lead to overfitting, especially with high-dimensional data, and can also be computationally expensive.

3.2 PCA-Based Model

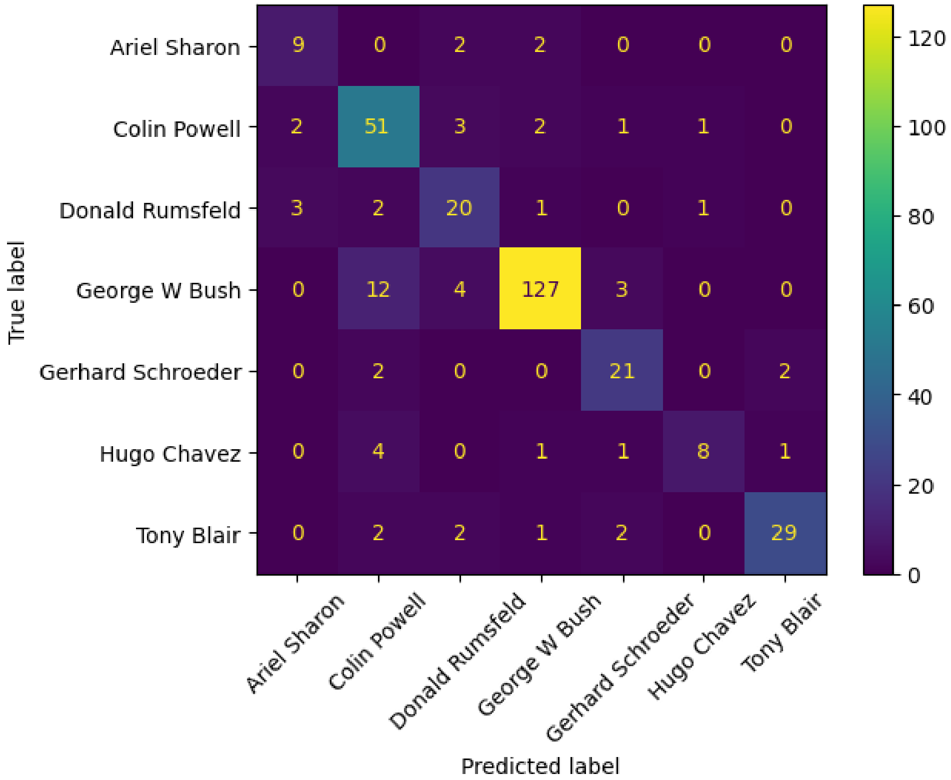

Figure 7: Confusion Matrix for PCA.

Performance Metrics:

- Accuracy: 82%
- Weighted Average Precision: 83%
- Weighted Average Recall: 82%
- Weighted Average F1-Score: 82%

Analysis:

- The PCA-based model has a slightly lower accuracy (82%) compared to the baseline model.
- PCA reduces the dimensionality by capturing the most significant variance in the data, which can help in reducing noise and improving generalization. However, PCA does not consider class separability, which might explain the drop in performance.
- PCA is useful for reducing computational complexity while retaining most of the variance in the data, but it may not be optimal for classification tasks where class discrimination is critical.

3.3 LDA-Based Model

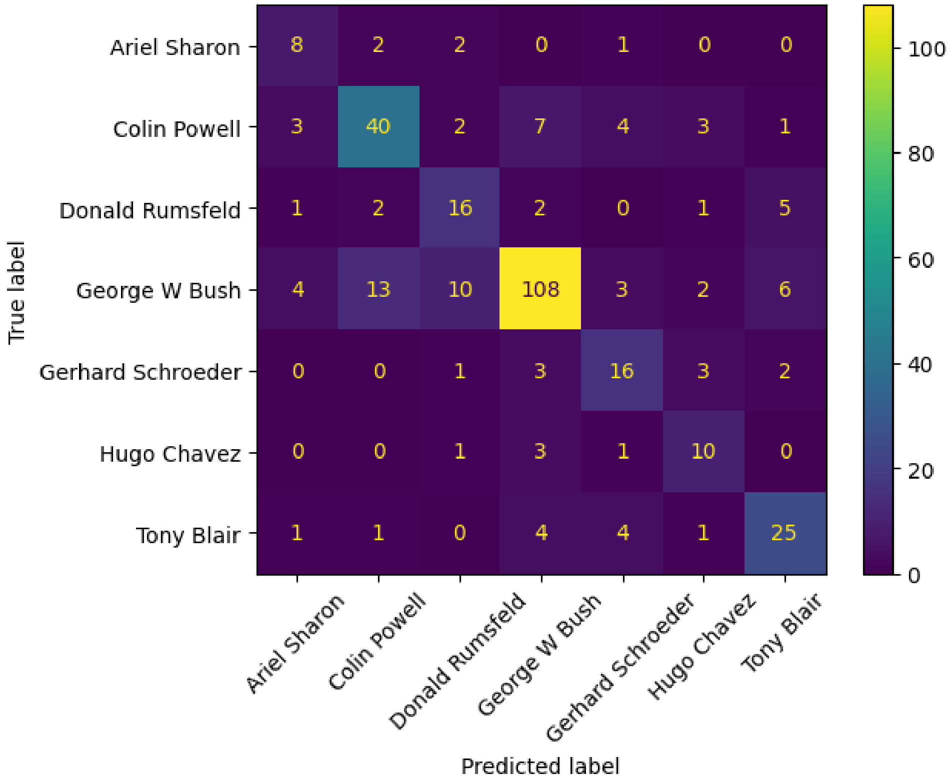

Figure 8: Confusion Matrix for LDA.

Performance Metrics:

- Accuracy: 69%
- Weighted Average Precision: 71%
- Weighted Average Recall: 69%
- Weighted Average F1-Score: 70%

Analysis:

- The LDA-based model has the lowest accuracy (69%) among the three models.
- LDA focuses on maximizing class separability, which is useful for classification. However, it may not always capture the most variance in the data, which could lead to a reduction in overall recognition performance.
- The lower performance could also be due to the limited number of discriminant components available (equal to min(n_classes − 1, n_features)), which might not capture enough information for complex datasets like LFW.

4 Draw Novel Insights in Comparison

4.1 Component Variation Analysis

4.1.1 PCA Component Variation:

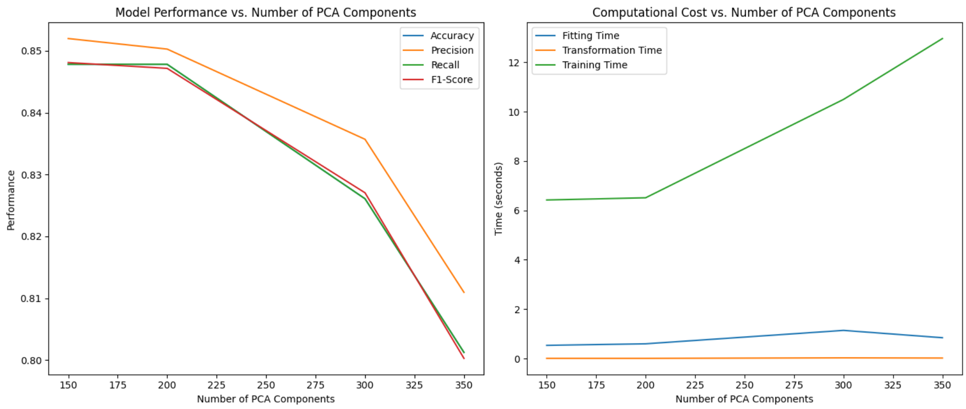

Figure 9: Model Performance and Computational Cost vs. Number of PCA Components.

- **Performance Metrics:**
  - Performance improves up to 200 components, with accuracy, precision, recall, and F1-score peaking.
  - Beyond 200 components, performance declines, indicating diminishing returns as more components add noise rather than useful information.
- **Computational Costs:**
  - Fitting and training times increase significantly with more components, reflecting higher computational demand.
  - Transformation time remains relatively stable, indicating consistent processing once the model is fitted.

4.1.2 LDA Component Variation:

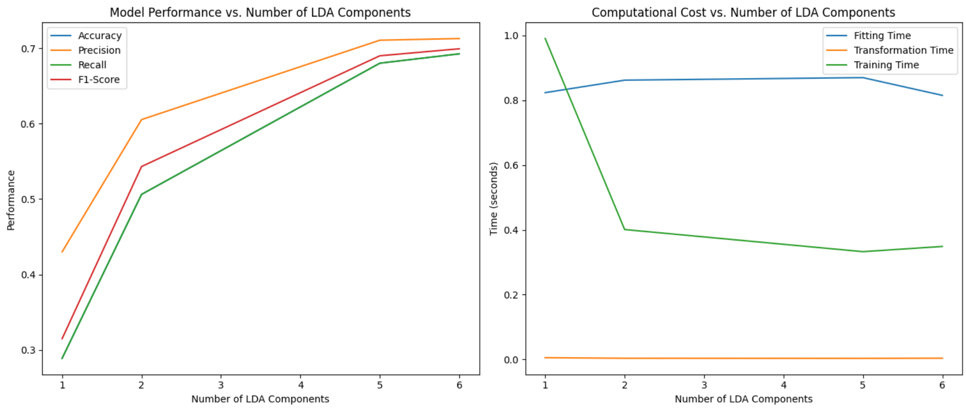

Figure 10: Model Performance and Computational Cost vs. Number of LDA Components.

- **Performance Metrics:**
  - Steady improvement with more components, as LDA optimizes class separability.
  - Performance metrics such as accuracy, precision, recall, and F1-score increase with more discriminant components.
- **Computational Costs:**
  - Fitting and training times decrease with more components, showing improved efficiency in feature extraction.
  - Transformation time remains relatively stable, indicating consistent processing once the model is fitted.

4.1.3 Observations

- **Less Training Time with More LDA Components:**
  - **Why This Happens**: Fewer LDA components may initially reduce dimensionality but may not provide optimal class separation, leading to longer training times. More components improve separability and reduce complexity.
  - **Difference from PCA**: LDA focuses on maximizing class separability, unlike PCA, which ignores class labels. This can lead to faster training with suitable components.
- **Conditions Affecting Computational Time:**
  - **Dataset Characteristics**: More components improve separability in overlapping classes, reducing training time.
  - **Classifier Choice**: Some classifiers (e.g., SVM, KNN) benefit more from feature reduction.

  - **Number of Classes**: LDA can only reduce dimensions to $c - 1$ components, where c is the number of classes.

- **Conclusion:** More LDA components often reduce training time by enhancing class separability. However, this isn't universally true and depends on factors like class separability and the classifier used. LDA's focus on class separability distinguishes it from PCA, affecting computational trends.

4.2 Insights on Model Performance

- **Dimensionality Reduction Impact:**

  - **PCA:** Captures overall variance. Optimal performance is achieved with an appropriate number of components; too many components can degrade performance due to noise.
  - **LDA:** Focuses on maximizing class separability, making it ideal for classification tasks. More components generally improve performance metrics.

- **Computational Efficiency:**

  - **PCA:** Becomes computationally expensive with high component counts. A balance is needed to optimize both performance and efficiency.
  - **LDA:** Generally more computationally efficient for fewer components and enhances class discrimination, making it suitable for classification.

- **Key Reasons:**

  - **LDA's Dimensionality Reduction:** LDA reduces the feature space to $\min(n_{\text{classes}} - 1, n_{\text{features}})$. With 7 classes in the dataset, LDA reduces 1850 features to just 6. This drastic reduction means LDA only keeps the most discriminative components, potentially losing much of the variance and important information, leading to underfitting.
  - **PCA's Approach:** PCA retains more components (e.g., 200), capturing more variance from the data. This provides a richer representation, preserving more information and allowing the model to better fit the data's complexity. By keeping more dimensions, PCA avoids underfitting, resulting in better classification performance.

- **Conclusion:** While LDA is theoretically better for class separation, it can suffer from underfitting when too few features are retained. This was evident in our analysis, where LDA sometimes failed to capture sufficient data complexity, especially with high-dimensional datasets like image classification tasks (e.g., the LFW People dataset). Conversely, PCA retains more components, striking a balance between dimensionality reduction and preserving variance, leading to higher accuracy, particularly in tasks with correlated features.

- **Recommendations and Applications:**

  - **Real-World Problems:**

    * In tasks like face recognition, where dimensionality is high, PCA's ability to retain variance across components makes it more effective than LDA, which risks underfitting.
    * In problems with clearer class separation (e.g., fraud detection or medical diagnosis), LDA is advantageous, focusing on distinguishing classes with fewer components.
  - **Improving Model Performance:**
    * Hybrid Approach: Applying PCA to reduce dimensionality, followed by LDA for class separation, works well for tasks like handwriting recognition, enhancing both variance preservation and class separability.
    * Feature Engineering: For tasks like spam detection, feature engineering alongside PCA can remove irrelevant features while maintaining core data structure.
  - **Practical Consideration:**
    * In Big Data scenarios, both PCA and LDA can become computationally expensive. Approximations like Incremental PCA or Online LDA can reduce this burden.
    * In deep learning, PCA is often used as a pre-processing step in models like CNNs to speed up training by reducing input dimensionality.

References

- Wikipedia contributors. (2024). *Linear discriminant analysis*. In *Wikipedia, The Free Encyclopedia*. Retrieved from `https://en.wikipedia.org/wiki/Linear_discriminant_analysis`

- Scikit-learn. (2024). *Comparison of LDA and PCA 2D projection of Iris dataset*. Retrieved from `https://scikit-learn.org/stable/auto_examples/decomposition/plot_pca_vs_lda.html`

- Scikit-learn. (2024). *LinearDiscriminantAnalysis*. Retrieved from `https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html`

- Martinez, A. (2011). *Fisherfaces*. Scholarpedia, 6(2), p. 4282. doi: 10.4249/scholarpedia.4282. Retrieved from `http://www.scholarpedia.org/article/Fisherfaces`

- Seymur. (2024). *Face Recognition (Eigenfaces, Fisherfaces, LBPH)*. Medium, 22 March. Retrieved from `https://medium.com/@seymurqribov05/face-recognition-eigenfaces-fisherfaces-lbph-0b39d41bd54c`

- Data Headhunters (2024). *PCA vs LDA: Dimensionality Reduction Techniques Explored*, Data Headhunters, 4 January. Retrieved from `https://dataheadhunters.com/academy/pca-vs-lda-dimensionality-reduction-techniques-explored/`

Code Repository

The code used in this report can be accessed by clicking the image below:

Click on the image to open the Google Colab notebook.