

# Model Development Phase Report

---

## Model Selection Report

For this project, the chosen approach combines Principal Component Analysis (PCA) for dimensionality reduction with an Artificial Neural Network (ANN) classifier. Linear Discriminant Analysis (LDA) was also applied after PCA to further maximize class separability.

## Model Descriptions

Model 1: PCA + ANN

Description: Baseline model using PCA eigenfaces as input features, classified with a Multi-layer Perceptron (MLP).

Model 2: PCA + LDA + ANN

Description: Enhanced model where PCA reduces dimensionality, LDA improves class separation, and ANN performs classification. This achieved higher accuracy compared to the baseline.

```

▶ # Step 4: PCA (Eigenfaces)
# =====
n_components = 150
print(f"Extracting the top {n_components} eigenfaces from {X_train.shape[0]} faces")

pca = PCA(n_components=n_components, svd_solver='randomized', whiten=True).fit(X_train)
eigenfaces = pca.components_.reshape((n_components, h, w))

eigenface_titles = [f"eigenface {i}" for i in range(eigenfaces.shape[0])]
plot_gallery(eigenfaces, eigenface_titles, h, w)
plt.show()

print("Projecting the input data on the eigenfaces orthonormal basis")
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)
print("Train PCA shape:", X_train_pca.shape, "Test PCA shape:", X_test_pca.shape)

```

```

▶ # =====
# Step 5: LDA (Fisherfaces)
# =====
lda = LinearDiscriminantAnalysis()
lda.fit(X_train_pca, y_train)
X_train_lda = lda.transform(X_train_pca)
X_test_lda = lda.transform(X_test_pca)
print("LDA projection done...")

```

⇒ LDA projection done...

```

▶ # =====
# Step 6: Classifier (MLP)
# =====
clf = MLPClassifier(
    random_state=1,
    hidden_layer_sizes=(10, 10),
    max_iter=1000,
    verbose=True
).fit(X_train_lda, y_train)

print("Model weights:", [coef.shape for coef in clf.coefs_])

```

## Implementation Pipeline

1. Dataset Preparation: Images were converted to grayscale, resized to 300x300 pixels, and flattened.
2. PCA (Eigenfaces): Extracted the most significant eigenfaces to reduce feature space.
3. LDA (Fisherfaces): Projected PCA features to maximize class separability.
4. ANN (MLP): A neural network with hidden layers (10,10) was trained.
5. Evaluation: Accuracy was tested across PCA components ( $k = 10-300$ ).
6. Imposter Detection: Low-confidence predictions were flagged using a threshold.