# A Face Verification System

*lir23gk*

University of Sheffield

## Abstract

This report presents a face verification system designed to determine whether two images belong to the same individual. The system integrates Support Vector Machines (SVM) with an RBF kernel for classification [1], Principal Component Analysis (PCA) for dimensionality reduction [2], and horizontal flipping for data augmentation [3]. Hyperparameter tuning using GridSearchCV [4] achieved a validation accuracy of 73% and an evaluation accuracy of 63.9%, outperforming the baseline k-Nearest Neighbors (kNN) model, which achieved 56.3%. The report outlines the system pipeline, experimental setup, results, and potential improvements.

## 1. Introduction

Face verification determines whether two images depict the same individual and is critical in applications such as biometric authentication, access control, and surveillance. Unlike face recognition, which identifies individuals from a database, face verification answers a binary question: "Are these two images of the same person?" This task is challenging due to high-dimensional image data, class imbalance, and the need for generalization to unseen samples [3].

This report details a face verification system combining Support Vector Machines (SVM) [1] with an RBF kernel, Principal Component Analysis (PCA) for dimensionality reduction [2], and horizontal flipping for data augmentation [3]. Hyperparameter tuning using GridSearchCV [4] optimized the system's performance. Key steps, experimental results, and areas for improvement are discussed.

## 2. System Description

The face verification system follows a structured machine learning pipeline integrating preprocessing, dimensionality reduction, and classification to address the challenges of high-dimensional data and class imbalance.

### 2.1. Preprocessing

The preprocessing stage includes:

- **Normalization:** Pixel intensities are scaled to the range [0, 1], ensuring consistent feature contributions.
- **Data Augmentation:** Horizontal flipping is applied to augment the dataset, doubling its size and mitigating class imbalance while enhancing generalization [3].

### 2.2. Dimensionality Reduction

Principal Component Analysis (PCA) reduces the dimensionality of the input data from 5828 features per image pair to 200 components, retaining over 95% of the variance. This step improves computational efficiency, reduces noise, and minimizes overfitting [2].

### 2.3. Classification

Support Vector Machines (SVM) with an RBF kernel are used to classify the reduced features into binary labels ("same" or "different"). The RBF kernel enables the model to capture non-linear decision boundaries, critical for distinguishing between the two classes [1].

### 2.4. Hyperparameter Tuning

GridSearchCV [4], with 3-fold cross-validation, is employed to optimize the following hyperparameters:

- $C$: Controls the trade-off between model complexity and margin maximization ($\{1, 10, 100\}$).
- $\gamma$: Defines the influence of individual samples in the RBF kernel ($\{$scale, 0.01, 0.1$\}$).

### 2.5. Pipeline Integration

The system is implemented as a pipeline that sequentially applies:

1. **StandardScaler:** Normalizes features to zero mean and unit variance.
2. **PCA:** Reduces dimensionality to 200 components.
3. **SVM:** Performs binary classification.

This modular structure ensures consistency, scalability, and reproducibility across all stages.

## 3. Experiments

This section summarizes the experiments conducted to optimize the face verification system, focusing on data preparation, hyperparameter tuning, and size compliance.

### 3.1. Training Data Selection

The dataset consists of image pairs labeled as "same" or "different." To address class imbalance and enhance diversity:

- **Data Augmentation:** Horizontal flipping was applied, doubling the dataset size and improving generalization to unseen data [3].
- **Train-Validation Split:** The augmented data was divided into 80% for training and 20% for validation.

### 3.2. Hyperparameter Optimization

GridSearchCV with 3-fold cross-validation [4] was used to tune SVM parameters:

- **Regularization ($C$):** Explored $\{1, 10, 100\}$ to balance model complexity.
- **Kernel Coefficient ($\gamma$):** Tested $\{$scale, 0.01, 0.1$\}$ to optimize the RBF kernel's flexibility.

The best configuration ($C = 10, \gamma =$ scale) achieved a validation accuracy of 73%.

### 3.3. Model Size Constraints

PCA reduced the feature dimensionality from 5828 to 200 components, significantly decreasing computational load [2]. The final trained model, including preprocessing and classification, occupied only 9.8MB, well within the 80MB limit.

### 3.4. Baseline Comparison

The baseline k-Nearest Neighbors (kNN) model achieved an evaluation accuracy of 56.3%. In contrast, the optimized SVM model with PCA and data augmentation improved the accuracy to 63.9%, demonstrating a substantial performance gain.

### 3.5. Evaluation Metrics

Validation accuracy was the primary metric for tuning, supported by:

- **Confusion Matrix:** Provided a breakdown of true/false positives and negatives.
- **Precision, Recall, and F1-Score:** Ensured balanced performance across classes.
- **ROC Curve and AUC:** The model achieved an AUC of 0.81, indicating robust classification performance [5].

## 4. Results and Analysis

This section highlights the system's performance on the evaluation dataset and identifies areas for improvement.

### 4.1. Evaluation Accuracy

The optimized system achieved an evaluation accuracy of 63.9%, outperforming the baseline k-Nearest Neighbors (kNN) model's 56.3%. This demonstrates the effectiveness of combining Support Vector Machines (SVM) [1], Principal Component Analysis (PCA) [2], and data augmentation [3].

### 4.2. Confusion Matrix and Metrics

The confusion matrix in Table 1 summarizes the classification results. The system showed slightly higher false negatives, indicating a tendency to misclassify "same" pairs. Table 2 provides the precision, recall, and F1-scores, with balanced performance across both classes.

Table 1: *Validation Confusion Matrix*

| | Predicted: 0 | Predicted: 1 |
|---|---|---|
| Actual: 0 | 323 | 116 |
| Actual: 1 | 119 | 322 |

Table 2: *Validation Metrics Summary*

| Class | Precision | Recall | F1-Score |
|---|---|---|---|
| 0 (Different) | 0.73 | 0.74 | 0.73 |
| 1 (Same) | 0.74 | 0.73 | 0.73 |

### 4.3. ROC Curve and AUC

The system achieved an AUC of 0.81, reflecting strong discrimination between "same" and "different" pairs. Figure 1 illustrates the trade-off between true positive and false positive rates.

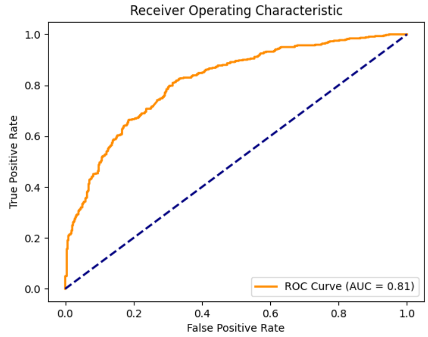


Figure 1: *ROC Curve of the Face Verification System with an AUC of 0.81.[5]*

### 4.4. Error Analysis

Misclassifications occurred mainly in the following cases:

- **False Negatives:** Often caused by subtle variations in lighting or facial expressions.
- **False Positives:** Observed when "different" pairs shared similar features, such as pose or background.

## 5. Discussion and Conclusions

The face verification system achieved an evaluation accuracy of 63.9%, significantly outperforming the baseline kNN model's 56.3%. The combination of SVM with an RBF kernel and PCA for dimensionality reduction effectively handled the challenges posed by high-dimensional data, while horizontal flipping enhanced data balance and model generalization.

Hyperparameter tuning using GridSearchCV proved instrumental in optimizing performance, with the model showing sensitivity to variations in lighting and expressions. Despite these advancements, false negatives highlight areas for improvement, such as advanced data augmentation or ensemble methods to enhance robustness.

The system's modular design underscored the importance of preprocessing and dimensionality reduction, while staying well within size constraints. Future work could explore class balancing techniques, such as SMOTE, to further mitigate misclassification issues and improve overall accuracy.

## 6. References

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[2] I. T. Jolliffe, *Principal Component Analysis*, second edition ed. New York: Springer, 2002.

[3] P. Y. Simard, D. Steinkraus, and J. C. Platt, "Best practices for convolutional neural networks applied to visual document analysis," in *International Conference on Document Analysis and Recognition (ICDAR)*, 2003, pp. 958–963.

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[5] T. Fawcett, "An introduction to roc analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.