Face Verification for Passport Ownership

Aqsa Shabbir
Department of Computer Engineering
Bilkent University
Ankara, Turkey
aqsa.shabbir@bilkent.edu.tr

Kousar Kousar
Department of Computer Engineering
Bilkent University
Ankara, Turkey
kousar.kousar@bilkent.edu.tr

Utku Oktay
Department of Computer Engineering
Bilkent University
Ankara, Turkey
utku.oktay@bilkent.edu.tr

***Abstract*—This work introduces a comprehensive system for automated passport ownership verification by comparing the claimant's facial features with the photograph on the passport personalization page. The system employs a custom implementation of the Viola-Jones (VJ) algorithm for robust face detection and Active Appearance Models (AAMs) for precise facial feature alignment. To address the challenges posed by variations in pose, lighting, and occlusions, we incorporate Eigenfaces, which utilize Principal Component Analysis (PCA) for dimensionality reduction and enhanced feature matching. Our methodology is evaluated using diverse datasets, including Yearbook, WIDER FACE, ATT Faces, and MUCT, achieving 97% accuracy in face detection, 0.01 RMSE from AAM in controlled scenarios and 83% F1 score for eigenfaces. All components of the system are implemented from scratch, ensuring adaptability and scalability. Our implementation is available at https://github.com/Kousarsaleem32/CS554_Project.**

***Index Terms*—Identity verification, Face localization, Passport Verification**

I. Introduction

The growing issue of identity fraud and document forgery necessitates robust tools for automated identity verification. Manual checks of official documents, like passports, are often slow and error-prone, leading to vulnerabilities in security systems. To address this, we propose a system that verifies passports by comparing the presenter's face with the photo on the passport's personalization page.

Our approach combines the Viola-Jones (VJ) algorithm for rapid face detection, Active Appearance Models (AAMs) for precise facial feature alignment, and Eigenfaces for enhanced face verification. VJ [1] provides robust and efficient face localization, making it an ideal first step for detecting the presenter's face. AAM [2] enables detailed alignment and feature extraction by modeling both the shape and texture of facial features. However, AAMs can be sensitive to variations in pose, lighting, and occlusions, which can hinder their effectiveness in challenging scenarios. To overcome these limitations, we incorporate Eigenfaces, which use Principal Component Analysis (PCA) for dimensionality reduction and capture essential facial features. This provides a stable and reliable approach for face verification, especially in cases where AAMs may struggle.

By integrating these techniques, our system effectively handles challenges such as lighting variations, pose differences, and occlusions, forming a cohesive and robust identity verification framework. While designed for passport verification, this approach is adaptable to other identity verification tasks, such as driver's licenses and national ID cards, offering a practical solution to enhance security and accuracy.

II. Related Work

Traditional computer vision techniques have been widely used for face detection, facial landmark detection, and face verification, offering robust, interpretable, and efficient solutions.

A. Face Detection

Face detection serves as a critical preprocessing step in face verification systems. The V&J framework [1], leveraging Haar-like features and AdaBoost, established a foundation for real-time applications with high detection accuracy. Enhancements such as optimized multi-classifier cascades [3] and hybrid methods for interactive systems [4] improved detection rates and processing speeds. Techniques like histogram equalization [5] address challenges like varying illumination, ensuring consistent feature extraction.

B. Facial Landmark Detection

Facial landmark detection identifies key points (e.g., eyes, nose, mouth) for alignment and feature extraction. AAM [2] integrates shape and texture via PCA, enable accurate landmark localization under moderate variations. Extensions to real-time applications [6] and methods like Constrained Local Models (CLM) [7] improve robustness using global shape constraints and local appearance models.

C. Face Verification

Face verification involves comparing features to confirm identity, using geometric metrics (e.g., inter-ocular distance) and texture descriptors like LBP and Haar-like features [3], [4]. These approaches, combined with preprocessing techniques [8], [5] enhance robustness to illumination and pose variations. Applications include simultaneous face detection and expression recognition [6] demonstrating versatility across domains.

III. Methodology

Our system involves three main stages: 1) face detection, 2) facial feature alignment, and 3) feature matching.

A. Face Detection

V&J algorithm is used to detect and extract face regions from passport pages. The methodology follows a structured approach, leveraging preprocessing, integral images, feature extraction, AdaBoost, and a cascaded classifier can be seen in Algo 1. Initially, positive and negative samples are loaded in grayscale format and resized to a uniform 24×24 pixel size for consistency. Integral images are then computed to enable rapid calculation of Haar-like features, where each pixel in the integral image represents the sum of all pixel values above and to the left in the original image.

Feature vectors are constructed by flattening the extracted features, and these vectors form the basis for training classifiers. The AdaBoost algorithm is employed to create a strong classifier by combining multiple weak classifiers, known as decision stumps. Each stump predicts based on a single feature, threshold, and polarity, and its weight is determined by the classification error. During training, sample weights are iteratively updated to focus more on misclassified examples, ensuring robust learning.

The cascade classifier is then trained in stages, where each stage uses an increasing number of features (5, 10 and 20 features) to train an AdaBoost model. Early rejection is employed in the cascade, where samples classified as non-faces in any stage are immediately discarded, reducing computational overhead.

For prediction, the cascade classifier processes input samples through each stage sequentially. If a sample is rejected at any stage, it is classified as a non-face. Only samples that pass all stages are considered as faces.

Algorithm 1 Algorithm for Viola and Jones

```
function PIPELINE
  DATASET ← Preprocess()
  Feat ← Extract()
  CASCADE ← Train()
  Evaluate(CASCADE, DATASET)
end function
function PREPROCESS
  images ← Load()
  for all image in images do
    Preprocess(image)
  end for
end function
function EXTRACT(DATASET)
  Feat ← []
  for all image in DATASET do
    haar ← ExtractHaar(image)
    Feat ← append(Feat, haar)
  end for
end function
function TRAIN(Feat)
  cascade ← InitCascade()
  for stage in stages do
    AdaBoost(cascade, Feat)
  end for
end function
function ADABOOST(Feat)
  for t in classifiers do
    TrainWeak(Feat)
    UpdateWeights()
  end for
end function
function EVALUATE(CASCADE, DATASET)
  for all sample in DATASET do
    CheckStages(CASCADE, sample)
    CountCorrect()
  end for
end function=0
```

B. Face Verification

1) Face Landmark Localization: Once we obtain the location of the face, we can find the landmarks on it. For this, we implement the AAM model [9]. The AAM model has both texture and shape models to capture the variations in both. To be able to model the shape variations, we first align the facial images properly to remove the effects of translation, rotation and scale through generalized Procrustes alignment. Then, to capture the shape variations, we apply PCA [10] and obtain a mean shape as well as the components explaining the variations.

After obtaining the mean shape, we warp the face images onto that shape so that we remove all shape variations and the only variation between the faces becomes the texture information. We then apply PCA on the face images and obtain a mean face image and the components.

We fit the AAM on a given image using Lucas-Kanade approach [11]. We simply warp the given image in such a way that the difference between pixel intensities of the mean image obtained from PCA and the provided image is minimized. Once we find the optimal warp, we use it to determine how we should change the mean shape to move the landmarks into the appropriate locations. We plan to exploit some features that are highly invariant between different images of a person, but varies between different individuals such as inter-ocular distance.

However, our approach using Active Appearance Models (AAMs) for face landmark localization and alignment faced challenges in achieving robust performance. These challenges, primarily stemming from variations in appearance, such as pose and illumination variations necessitated a shift to a more efficient and scalable solution. This led us to adopt the eigenfaces approach, which leverages Principal Component Analysis (PCA) for feature extraction and dimensionality reduction.

2) Eigenfaces: Eigenfaces captures the most significant facial features while reducing computational complexity. During preprocessing, all images were converted to grayscale and resized to a uniform dimension. Each image was vectorized into a one-dimensional array to facilitate PCA application. PCA was employed to compute the eigenfaces, which represent the directions of maximum variance in the training set. These eigenfaces provided a lower-dimensional representation of the original images while preserving the most discriminative features. Each image was projected into the eigenspace defined by the principal components, resulting in a compact feature vector.

To classify these feature vectors, we explored various machine learning approaches. Pairwise comparisons between feature vectors were performed by subtracting one vector from another, generating a difference vector as input for the classifiers. The machine learning approaches evaluated included:

- **L2 Distance (Euclidean Distance):** Used to calculate the scalar distance between feature vectors.
- **Support Vector Machines (SVM):** A classification model for finding optimal hyperplanes separating the data.
- **Decision Tree:** A simple model for interpretable, rule-based classification.
- **Random Forest:** An ensemble learning technique combining multiple decision trees for improved performance.
- **XGBoost:** A gradient boosting algorithm that uses iterative training to enhance classification accuracy.
- **CatBoost:** Another gradient boosting algorithm that builds symmetric trees.

IV. Data Splits and Hyperparameters

A. Data Splits

- **V&J Face Detection:** A total of 10,000 examples were used, with an equal split of 5,000 positive examples (images with faces) from Yearbook Dataset [12] and 5,000 negative examples (images without faces) from WIDER FACE Dataset [13]. The dataset was divided into 80% training (8,000 images) and 20% testing (2,000 images).
- **AAM:** AAMs are trained using the MUCT Dataset [14] which includes detailed landmark annotations.
- **Eigenfaces:** We have employed AT&T Faces dataset [15], containing 40 individuals, each with 10 images featuring varying expressions. We have used 30 individuals for training, 4 for validation and 6 for testing.

V. Evaluation

A. Viola Jones Face Detection

The system achieved an accuracy of 97% in detecting faces on the test set, confirming its reliability for face localization tasks. The cascaded classifier structure used in the V&J method with an increasing number of features (5,10 and 20) plays a critical role in enabling high detection accuracy while maintaining computational efficiency. A sample visualization can be seen in figure 1.

Algorithm 2 Eigen Faces

Notation: T_r = Train set, T_e = Test set, C_r = Train components, C_e = Test components, X_r, y_r = Train features and labels, X_e, y_e = Test features and labels, n = Number of principal components retained in PCA, `mdl` = trained model, `(prec, rec)` = Precision and Recall.

function MAIN()
  $T_r, T_e \leftarrow$ LOADSPLIT(`data`)
  `pca`, $C_r, C_e \leftarrow$ APPLYPCA(T_r, T_e, n)
  $X_r, y_r \leftarrow$ EXTRACTFEAT(C_r)
  $X_e, y_e \leftarrow$ EXTRACTFEAT(C_e)
  `prec, rec` $\leftarrow$ TRAINEVAL(X_r, y_r, X_e, y_e)
  `pred` $\leftarrow$ PREDICT(`pca, pairs`)
end function
function LOADSPLIT(`Dataset`)
  `img` $\leftarrow$ `LoadImg(Dataset)`
  $T_r, T_e \leftarrow$ `img[:30], img[34:]`
end function
function APPLYPCA(T_r, T_e, n)
  `pca` $\leftarrow$ `FitPCA(`T_r, n`)`
end function
function EXTRACTFEAT(C)
  return `PairwiseFeat(`C`, Labels(`C`))`
end function
function TRAINEVAL(X_r, y_r, X_e, y_e)
  `mdl` $\leftarrow$ `TrainClf(`X_r, y_r`)`
  `pred` $\leftarrow$ `mdl.Predict(`X_e`)`
  return `Prec(`y_e`, pred), Rec(`y_e`, pred)`
end function
function PREDICT(`pca, pair`)
  return `Out(ProjectPair(pca, pair))`
end function=0

Fig. 1: Detected Faces by V&J

Despite the high accuracy, the algorithm's performance could be impacted by several factors, including occlusions, non-frontal views, and varying lighting conditions. While the test accuracy was high, real-world scenarios may present additional challenges that require further tuning of the system.

B. Active Appearance Models

The evaluation of the AAM approach revealed several challenges. Despite using annotated landmarks from the MUCT dataset, the fitting process often diverged under even moderate appearance variations. The AAM implementation achieved an

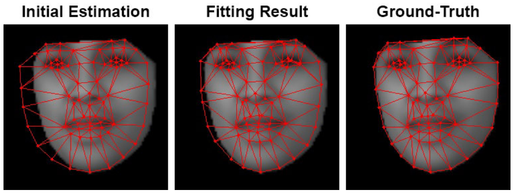

Fig. 2: A partially successful fitting example from our AAM implementation.

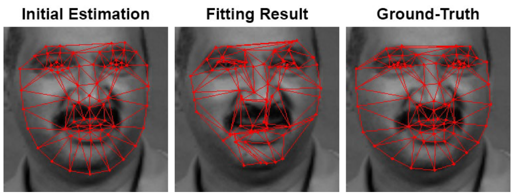

Fig. 3: An unsuccessful fitting example from our AAM implementation.

RMSE (root mean squared error) of around 0.01 in terms of normalized pixel values (the sum of the distances between each estimated landmark in normalized pixel values) for landmark localization in highly controlled conditions as depicted in Fig. 2, but rised to 0.04 and would diverge without maximum iteration number limit under some appearance variation as shown in Fig. 3. These limitations stemmed from difficulties in achieving stable convergence and the sensitivity of the model to resolution changes.

C. Eigen Faces

We have tested our model with various numbers of principal components (5, 10, and 20) to evaluate its impact on performance. Table I summarizes the results, showing improvements in precision, recall, and F1 score as the number of components increases. The best performance was achieved with 20 components, where precision, recall, and F1 score were 0.76, 0.94, and 0.84, respectively.

TABLE I: Performance metrics for different PCA components.

PCA Components	Precision	Recall	F1 Score
5	1.00	0.52	0.68
10	0.97	0.69	0.80
20	0.96	0.73	0.83

Beyond Euclidean distance, various machine learning classifiers were evaluated for their ability to distinguish between feature vectors derived from PCA. Table II summarizes the performance of these classifiers using 20 PCA components. Random Forest outperformed other methods, achieving an F1 score of 0.87. The other models did not bring a significant improvement. The hyperparameters regarding these results are given in Table III.

We finally tested our Random Forest model with the provided hyperparameters on the seperated test set, where we achieved a precision of 0.78, a recall of 0.90 and an F1 score of 0.83. The findings highlight the strength of combining PCA-based dimensionality reduction with robust classifiers like Random Forest. However, limitations remain, including performance discrepancies due to the dataset's small size and sensitivity to noise in the input data. While the Euclidean distance approach provided a baseline, ML classifiers significantly improved the verification process by capturing more complex patterns in the data.

TABLE II: Performance metrics of ML classifiers with 20 PCA components.

Classifier	Precision	Recall	F1 Score
Euclidean Distance	0.96	0.73	0.83
SVM	0.72	0.98	0.83
Decision Tree	0.83	0.78	0.80
Random Forest	0.96	0.80	0.87
XGBoost	0.82	0.86	0.84
CatBoost	0.82	0.81	0.82

TABLE III: Best hyperparameters of ML classifiers.

Model	Best Hyperparameters
SVM	kernel='rbf', gamma='scale', C=1
Decision Tree	min_samples_split=10, min_samples_leaf=8, max_depth=20, criterion='log_loss'
Random Forest	n_estimators=200, min_samples_split=10, min_samples_leaf=4, max_depth=20
XGBoost	subsample=0.6, n_estimators=50, max_depth=9, learning_rate=0.1, colsample_bytree=0.6
CatBoost	num_trees=1000, learning_rate=0.1, l2_leaf_reg=1, depth=6

VI. Conclusion

Our work demonstrates a custom-built system for passport ownership verification through V&J face detector and AAM model alignment techniques. Our feature matching and verification phases are still in progress. By implementing all components our system will offer a reliable and adaptable solution for secure identity verification.

References

[1] P. Viola and M. J. Jones, "Robust real-time face detection."
[2] T. F. Cootes and Edwards, "Active appearance models." Springer, 1998.
[3] L. Lang and W. Gu, "Study of face detection algorithm for real-time face detection system." IEEE, 2009.
[4] J. Zhu and Z. Chen, "Real time face detection system using adaboost and haar-like features." IEEE, 2015.
[5] M. Phankokkruad and P. Jaturawat, "An evaluation of technical study and performance for real-time face detection." IEEE, 2015.
[6] M. S. Bartlett, G. Littlewort, I. Fasel, and J. R. Movellan, "Real time face detection and facial expression recognition." IEEE, 2003.
[7] D. Cristinacce, T. F. Cootes *et al.*, "Feature detection and tracking with constrained local models." Edinburgh, 2006.
[8] G. Kukharev and A. Nowosielski, "Visitor identification-elaborating real time face recognition system," 2004.
[9] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models."
[10] K. Pearson, *The London, Edinburgh, and Dublin philosophical magazine and journal of science*.
[11] S. Baker and I. Matthews, "Lucas-kanade 20 years on: A unifying framework."
[12] S. Ginosar and Rakelly, "A century of portraits," 2015.
[13] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," 2016.
[14] S. Milborrow, "The muct landmarked face database," *Pattern Recognition Association of South Africa*, 2010.
[15] F. Wu, Q. Xiao, and T. D. Vo, "Face image database," *International journal of biometrics*, 2013.