FPT UNIVERSITY

FPT UNIVERSITY HCMC

Face Verification Using SVM, LBP, PCA, and Face Cascade

Computer Vision - SU24

Le Thanh Tha — SE182357

1. Introduction

Face verification is one of the most important task which can predict person identity by their face's images. For a few decade, with the grow of the high-tech devices we need method more secure for our smart devices like phone and laptop which contain numerous of essential data than PIN, password or OTP which can bypass by hacker.

In this report we will exploring machine learning method for face verification by combine these method Support Vector Machine (SVM)[1] for classification, Local Binary Patterns (LBP) for feature extraction [5], Principal Component Analysis (PCA) for dimensional reduction [3], and Haar Cascade Classifiers for face detection [6].

2. Problem Definition

Face verification for personal identify is essential security for authentication applications which provide a non-intrusive and user-friendly way thank for comparing live face's images from webcam to face's images from dataset. I suppose that utilize machine learning method can assist we to build real-time and optimize hardware used for that authentication task.
My facial verification method will divided into three parts:

- **Face Registration**: The client will enter their username, and the system captures images of their face using a webcam to store in the dataset.
- **Model Training**: Load all face images of the clients for extracting characteristics, reducing characteristics, and training the dataset.
- **Face Verification**: The client will enter their username, and the system will capture images of the client's face using a webcam, taking frames one by one. These frames will be predicted and compared to the entered username. Verification only complete when the true predict username matches the entered username in more than 60% of the frames from a sequence of 30 frames.

3. Method

3.1. Face Registration

Face detection is performed using OpenCV's Haar Cascade Classifier (`face_cascade`), which detects faces in real-time from the webcam (Figure 1). During registration, users enter their username, and the system captures and saves multiple images of their face.

3.2. Feature Extraction

LBP (Local Binary Patterns) extracts features from face images by analyzing the local texture of the face (Figure 2). This helps capture the face's structure and appearance [5].

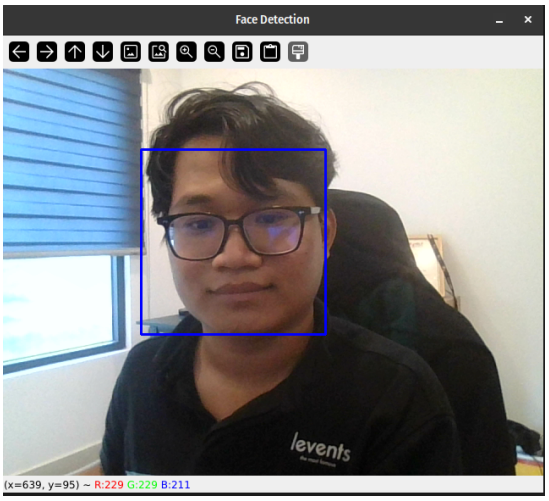


Figura 1: Face detection using Haar Cascade Classifier

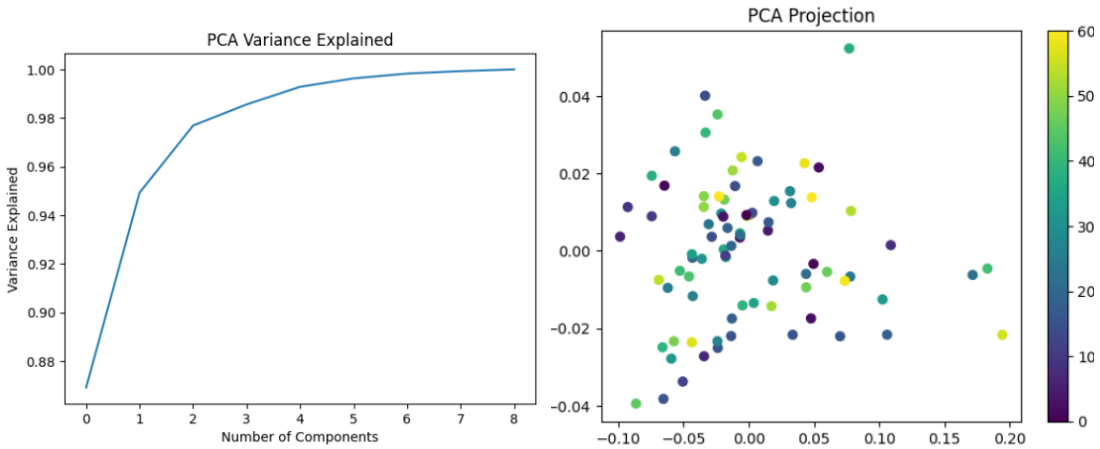
Figura 2: LBP feature extraction

3.3. Dimensionality Reduction

PCA (Principal Component Analysis) is then applied to reduce the size of the LBP features, keeping the most important information while making it easier to process.

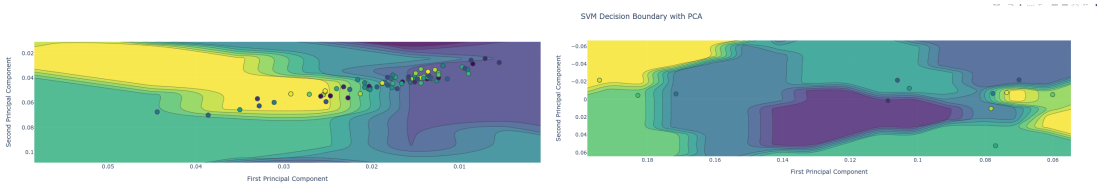
3.4. Model Training

SVM is trained using the LBP features after PCA reduces their dimensionality. SVM is effective at handling high-dimensional data and works well for face recognition tasks [1].



(a) PCA Variance (b) PCA Projection

Figura 3: PCA feature reduction visualize



(a) SVM without PCA (b) SVM with PCA

Figura 4: SVM visualize

4. Implementation and Results

4.1. Implementation

4.1.1. Dataset

The dataset consists of face images collected during the registration phase and additional images from the Labeled Faces in the Wild (LFW) dataset[4]. Specifically, 1% of the LFW dataset is used, combined with the registered face data. All images are saved in grayscale, and face images are detected and captured using the Haar Cascade classifier to ensure consistency and improve the robustness of the face verification system [2].

4.1.2. Face Registration

1. Initialize the webcam and prompt the user to enter their username.

2. Use OpenCV's Haar Cascade Classifier to detect faces in the webcam feed.

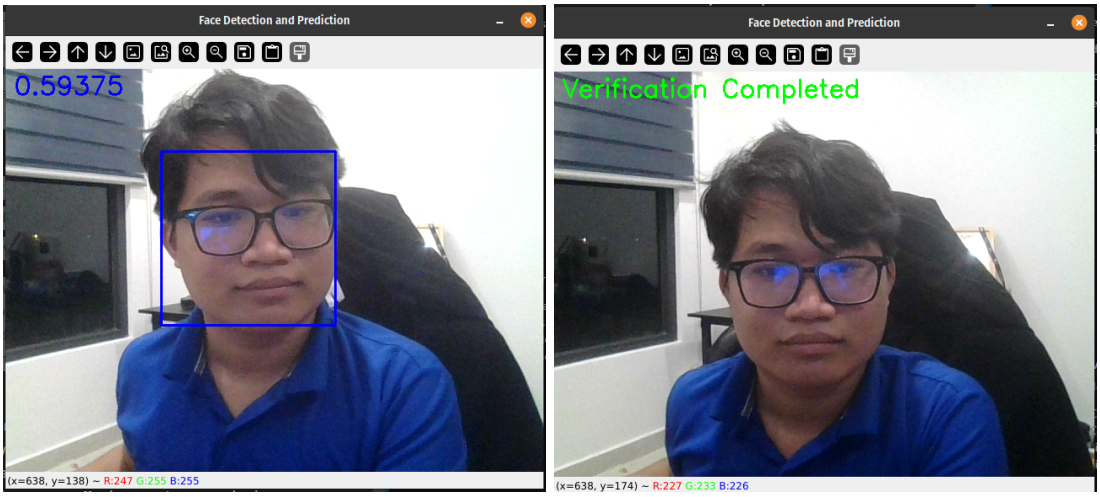
3. Capture multiple images of the user's face and save them in grayscale format in a dataset directory under the entered username.

4.1.3. Model Training

1. Load the collected face images and the Labeled Faces in the Wild (LFW) dataset (1% sample).
2. Convert all images to grayscale (if not already).
3. Use LBP to extract features from each face image.
4. Apply PCA to reduce the dimensionality of the LBP features.
5. Train an SVM classifier using the processed features and corresponding usernames as labels.

4.1.4. Face Verification

1. Initialize the webcam and prompt the user to enter their username.
2. Use OpenCV's Haar Cascade Classifier to detect the face in the webcam feed.
3. Capture the face image and convert it to grayscale.
4. Extract features from the captured face image using LBP.
5. Apply PCA to the extracted LBP features.
6. Use the trained SVM model to predict the username based on the processed features.
7. Compare the predicted username with the entered username. If the predicted username matches the entered username in more than 60% of the frames from 30 consecutive frames, the verification is considered complete.



(a) Proccessing face verification (b) Face verification complete

Figura 5: Face verification process

4.2. Results

4.2.1. Experimental Results

The Haar Cascade Classifier showed good results in terms of face detection. LBP operated correctly with regard to feature facial, and PCA has been successfully implemented and as a result has decreased the number of features while preserving the necessary information. The SVM model was evaluated based on several metrics, including accuracy and F1-score:

1. Accuracy: 0.41
2. F1-Score: 0.33

The execute time was shorter than that of the deep learning techniques, thus, making the method appropriate for real-time applications. Although, with the rise in the number of images, the accuracy of the identification was low and the training time was long. This limitation is usually observed when working with big data sets in support vector machine.

4.2.2. Conclusion

In my report, my face verification system using SVM, LBP, PCA and Haar Cascade. The outcome of the face verification is possible by feature extraction with LBP, followed by coming down in the dimension with PCA and final classification with SVM. The results of testing also support the effectiveness of the given system and state the possibility of its application in real-time processes connected to security and access control for limited hardware and small authentication application.

References

[1] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. pages 2, 3

[2] Gary B Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. *University of Massachusetts, Amherst, Technical Report*, 2008. pages 4

[3] Ian T Jolliffe. *Principal Component Analysis*. Springer Series in Statistics, 2002. pages 2

[4] Martin Knoche, Stefan Hormann, and Gerhard Rigoll. Cross-quality lfw: A database for analyzing cross- resolution image face recognition in unconstrained environments. In *2021 16th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2021)*, pages 1–5, 2021. pages 4

[5] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002. pages 2

[6] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. IEEE, 2001. pages 2