Computer Vision Project – Face Recognition

Exercise 4.1 – 4.5 Report

Exercise 4.1 – Face Detection, Tracking, and Alignment

1. Limitations of Template-Based Tracking

To reduce computational cost, template matching is used to track faces between frames after an initial detection. While efficient, this approach exhibits several limitations:

Template matching is sensitive to large head pose variations, as strong rotations significantly change facial appearance.

Occlusions and illumination changes can reduce similarity scores and lead to tracking failures.

Scale changes cannot be handled, since template matching assumes a fixed face size. When a subject moves closer to or farther from the camera, tracking accuracy decreases.

Fast motion may cause the face to leave the predefined search window, resulting in loss of tracking.

Background regions with similar appearance may lead to false matches.

Due to these limitations, periodic re-detection using a face detector is required to maintain tracking robustness.

2. Parameter Tuning

Several parameters influence the performance of the tracking system:

A larger search window improves robustness to motion but increases computational cost and the likelihood of false matches.

A higher similarity threshold triggers re-detection more frequently, improving accuracy but reducing efficiency.

A lower similarity threshold increases the risk of template drift over time.

All parameters were tuned experimentally to balance robustness and computational efficiency.

Exercise 4.2 – Face Identification

1. Closed-Set Face Identification

Face identification is performed using deep face embeddings and a k-nearest neighbor (k-NN) classifier in embedding space. Training data is collected either from a webcam or from the provided dataset, and embeddings are stored in a gallery together with their labels.

During testing, the identity of a detected face is determined by comparing its embedding to the gallery using Euclidean distance.

2. Influence of the Parameter k

The number of neighbors k in k-NN classification has a significant impact on performance:

Experiments were conducted using different values of k (k = 1, 3, and 5).

For k = 1, the system is highly sensitive to noise, leading to unstable predictions across frames.

Increasing k improves robustness by aggregating multiple neighbors.

For k = 5, predictions are more stable, but class boundaries become less distinct.

3. Open-Set Identification

To evaluate open-set behavior, one identity is deliberately excluded from training. During testing, the system correctly assigns the label *unknown* to this identity based on distance or confidence thresholds. This demonstrates that the system is capable of rejecting identities that are not present in the gallery.

Exercise 4.3 – Face Clustering

Unsupervised face clustering is performed using k-means on face embeddings.

1. Convergence Analysis

The k-means objective function (sum of squared errors, SSE) is monitored over iterations. For different subjects, the SSE consistently decreases, indicating convergence. More challenging identities show higher initial SSE values, but the algorithm converges after a small number of iterations.

The results also show sensitivity to initialization, which is a known characteristic of k-means clustering.

2. Face Re-identification via Clustering

For re-identification, the embedding of a test face is compared to cluster centers, and the closest cluster is selected. This approach enables identity re-identification without explicit labels and works reliably for multiple subjects.

Exercise 4.4 – Evaluation of Face Recognition

Open-set recognition performance is evaluated using Detection and Identification Rate (DIR) curves. DIR measures the correct identification rate of known identities, while the False Alarm Rate (FAR) measures how often unknown identities are incorrectly accepted.

By varying the similarity threshold, a trade-off between identification accuracy and false alarms is observed. This evaluation enables informed threshold selection depending on application requirements.

Exercise 4.5 – Open-Set Face Recognition Challenge

1.Problem Setting

The challenge addresses open-set face recognition, where the training data contains Known Classes (KC) and Known Unknown Classes (KUC). During testing, additional Unknown Unknown Classes (UUC) may appear. The goal is to correctly identify KC while rejecting all unknown identities.

2. Feature Representation

The provided 128-dimensional face embeddings are used directly. No access to raw images or feature extraction is required, and the focus lies on open-set learning and decision strategies.

3. Single Pseudo Label (SPL)

In the Single Pseudo Label approach, all KUC samples are mapped to a single pseudo class during training. A multi-class classifier is trained on KC and this pseudo class.

During testing, samples predicted as the pseudo class are classified as *unknown*. Additionally, predictions with low confidence for KC are rejected based on a threshold.

4. Multi Pseudo Label (MPL)

The Multi Pseudo Label approach extends SPL by clustering KUC samples using k-means. Each cluster is treated as a separate pseudo class during training.

At test time, any prediction assigned to a pseudo class is mapped to *unknown*. This approach provides a finer modeling of the unknown space at the cost of increased complexity.

5. Open-Set Decision Rule

The maximum posterior probability over Known Classes is used as a known-score. A threshold is applied to this score to reject uncertain predictions. The threshold is calibrated on KUC samples to control the false alarm rate and balance robustness and accuracy.

6. Implementation and Efficiency

A linear classifier with standardized features is used as the backbone. SPL provides fast training and inference, while MPL uses MiniBatch k-means to maintain efficiency. All methods run within reasonable time and pass the provided test cases.

Conclusion

This project implements a complete face recognition pipeline with robust handling of unknown identities. The use of pseudo labeling enables effective open-set recognition, and experimental results demonstrate the stability and flexibility of the proposed methods across different scenarios.