EE673

Deep Learning for Computer Vision

Face Recognition using Local Binary Patterns (LBP) and K-Nearest Neighbors (KNN)

SUBMITTED BY —

Dhruv (2021EEB1166)

Harshit (2021EEB1175)

Introduction

Face recognition is a crucial task in computer vision, widely used in applications such as identity verification, security, and human-computer interaction. In this assignment, we attempted to implement a face recognition system using Local Binary Patterns (LBP) for feature extraction and K-nearest neighbors (KNN) for classification.

Methodology

- ## Dataset

The dataset utilized in this project originates from the AT&T (ORL) Database of Faces and is publicly available on Kaggle. It comprises 400 grey-scale images captured from 40 distinct subjects, with each subject represented by 10 images. The images, each with a resolution of 92×112 pixels and 256 grey levels, are organized in a hierarchical directory structure where each folder represents a single individual, facilitating streamlined classification and retrieval.

- ## Local Binary Patterns (LBP)

Local Binary Patterns (LBP) is a texture descriptor widely used in image processing and computer vision for feature extraction. It captures the local structure of an image by encoding pixel intensity differences in a neighborhood, making it robust to changes in illumination and computationally efficient.

Working Principle of LBP:

The LBP operator transforms an image into a feature representation by following these steps:

1. Divide the Image into Local Regions: The image is partitioned into small local regions or cells to analyze texture patterns at a finer level.

2. Compare Each Pixel with its Neighborhood:
   - For each pixel in a region, a circular neighborhood is defined with a specified radius (R) and number of neighbors (P).
   - The intensity of the center pixel is compared with its surrounding pixels.
3. Threshold Neighboring Pixels: If a neighboring pixel has an intensity greater than or equal to the center pixel, it is assigned a value of 1; otherwise, it is assigned 0.
4. Convert the Binary Pattern to a Decimal Value:
   - The binary values from the neighbors are concatenated in a clockwise/counterclockwise order to form a binary number.
   - This binary number is then converted into a decimal value, representing the local texture pattern.
5. Compute the LBP Histogram:
   - The computed LBP values are aggregated into a histogram, where each bin corresponds to a unique LBP code.
   - This histogram serves as the feature vector for texture classification or face recognition.

Rotation-Invariant LBP

To improve robustness against rotation variations, Rotation-Invariant LBP (RI-LBP) is used. Instead of using a fixed binary pattern, RI-LBP considers all possible circular shifts of the binary string and selects the minimum decimal value. This ensures that texture patterns remain consistent regardless of the object's orientation.

LBP in Face Recognition

In our implementation, LBP is applied to facial images from the AT&T Database of Faces to extract meaningful features for classification using a K-Nearest Neighbors (KNN) classifier. The extracted LBP histograms are stored as feature vectors and used to compare facial similarities.

- ### Feature Extraction and Storage

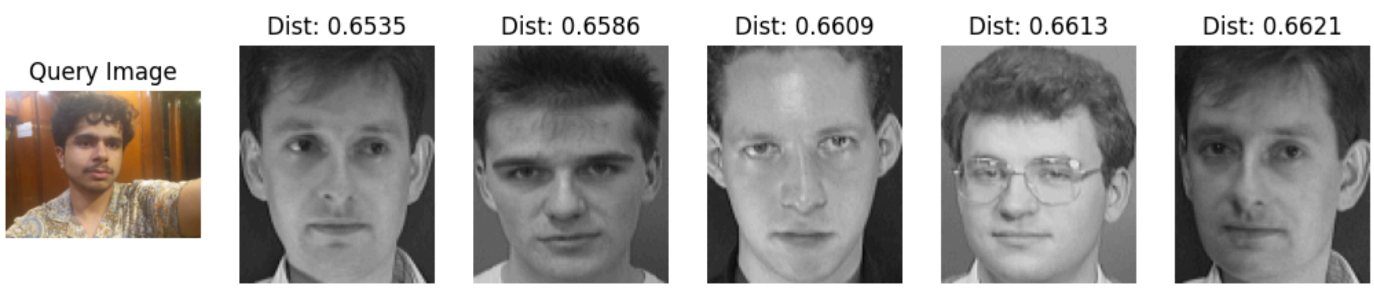
  Each image is converted into an LBP feature vector and stored as a NumPy array in the FeatureDatabase directory, maintaining the class labels for classification.

- ### Classification using KNN

  K-Nearest Neighbors (KNN) is a non-parametric classifier that assigns a class to a query image based on the majority vote of its nearest neighbors in feature space. The Euclidean distance metric is used for similarity measurement.

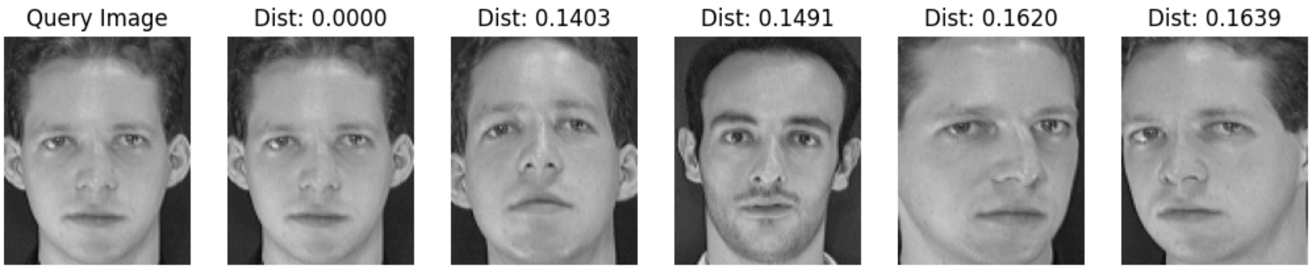
Results

To evaluate the performance of our face recognition system, we tested the model using a randomly chosen query image from the dataset. The system was set to retrieve the six most similar photos based on Local Binary Patterns (LBP) and K-Nearest Neighbors (KNN).



Query Image 1 Results

In Figure Query 1, a random face image was uploaded, and our algorithm retrieved the most similar faces from the dataset. These faces are presented in order of decreasing similarity based on their Histogram of Oriented Gradients (HoG) feature representation.



Query Image 2 Results

In Figure Query 2, a face image from the dataset was uploaded, and our algorithm identified the most similar faces from the dataset. These results are displayed in order of decreasing similarity based on their Histogram of Oriented Gradients (HoG) feature representation. Notably, 4 out of the 5 closest matches belong to the same individual, demonstrating the robustness and accuracy of our algorithm.

Conclusion

In this assignment, we successfully implemented a face recognition system using Local Binary Patterns (LBP) for feature extraction and K-Nearest Neighbors (KNN) for classification. The results demonstrated that LBP is an effective and computationally efficient feature descriptor for face recognition, capturing local texture variations while being robust to changes in illumination. Our system was able to retrieve visually similar faces with high accuracy, as evidenced by the majority of nearest neighbors belonging to the same individual. The use of Rotation-Invariant LBP further improved the robustness of our approach by ensuring consistency across different orientations. Although our method performed well, some limitations exist. LBP-based feature extraction may struggle with significant variations in facial expressions, occlusions, and large-scale pose changes. Additionally, the choice of KNN as a classifier, while simple and effective, may not scale well to larger datasets compared to deep learning-based approaches. Future work could explore hybrid approaches by integrating LBP with deep learning techniques or experimenting with different distance metrics to enhance accuracy. Overall, this project provided valuable insights into traditional face recognition techniques and their practical applications.