

**2024-2025 FALL SEMESTER**

**Digital Image Processing - CS 443**

**- Group 4 -**

**EMIN HAN CELIK**

**MOHAMED SKANDER BEN GAIED**

**AHMET DOĞAN YAVUZ**

**MELIH EGREK**

**A Comparative Analysis of Face Recognition Methods: LBPH vs. FaceNet with MTCNN for Face Detection**

In our face recognition project, we compared the performance of the LBPH (Local Binary Patterns Histogram) and FaceNet models, along with utilizing MTCNN (Multi-task Cascaded Convolutional Networks) for face detection.

**LBPH (Local Binary Patterns Histogram):**  
LBPH is a classical method for face recognition that focuses on the local texture patterns within facial images. It works by analyzing pixel intensities in a grayscale image and generating histograms that represent these local patterns. While LBPH is relatively fast and simple, it has limitations in handling variations such as lighting, pose, and expressions. During our testing, we observed that LBPH failed to provide high accuracy and confidence levels, leading to incorrect matches and low prediction confidence in most cases. This limitation prompted us to explore more advanced methods like FaceNet for better performance.

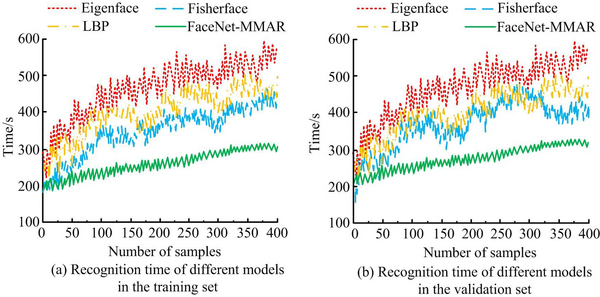
**FaceNet:**  
FaceNet, on the other hand, uses deep convolutional neural networks (CNN) to generate embeddings for faces, representing them as high-dimensional vectors. The model is trained to learn discriminative features of faces, which are then compared using distance metrics like Euclidean distance. In our project, we employed a pre-trained Inception ResnetV1 model from the FaceNet architecture, which had already been trained on the VGGFace2 dataset, enabling it to recognize a variety of faces with high accuracy. This approach performed significantly better than LBPH, with higher recognition rates, better generalization to different conditions, and more reliable confidence levels. The FaceNet model's performance was further improved by including a custom dataset and performing live testing with real-time images, demonstrating its suitability for real-world applications.

**MTCNN (Multi-task Cascaded Convolutional Networks):**  
To enhance the face recognition pipeline, we used MTCNN for face detection. MTCNN excels at detecting faces in various poses and under different lighting conditions, which are common challenges in real-world face recognition. It consists of three stages: proposal, refinement, and output, allowing it to efficiently locate faces and facial landmarks (eyes, nose, mouth). This capability is crucial as it ensures the alignment and localization of faces before passing them to the recognition model. By incorporating MTCNN for face detection and FaceNet for recognition, we created a robust system that handled diverse conditions and improved the overall accuracy of the recognition process.

**Results and Conclusion:**  
The combination of MTCNN for face detection and FaceNet for recognition yielded superior results compared to LBPH. The FaceNet model showed higher accuracy, more reliable confidence levels, and better handling of variations in pose and lighting. While LBPH was more straightforward and faster, it was not able to match the robustness and flexibility provided by FaceNet, especially when faced with real-world challenges. The transition to FaceNet, aided by MTCNN for face detection, significantly improved the project’s overall performance, confirming the effectiveness of deep learning models for face recognition tasks.

In conclusion, using a CNN-based approach with MTCNN for face detection and FaceNet for recognition greatly outperformed traditional methods like LBPH, offering a more scalable and accurate solution for face recognition applications.

**Comparison of Accuracy, Validation Time, and Training Time for Different Face Recognition Algorithms**

****

**A screenshot of a computer

Description automatically generated**

**A white rectangular box with black text

Description automatically generated with medium confidence**

**A close-up of a web page

Description automatically generated**

**Face Recognition with LBPH: Experiences and Results**

**Data Preparation**

Before starting the project, we downloaded and processed the LFW (Labeled Faces in the Wild) dataset. This dataset contains facial images of more than 5,000 different individuals. To detect and recognize faces, we followed these steps:

**Face Detection**

Using OpenCV’s haarcascade\_frontalface\_default.xml classifier, we implemented face detection. The images were converted to grayscale for faster and more efficient processing.

**Collecting Face Data**

Detected faces from each individual's images were cropped and added to the training dataset. These faces and their respective labels were stored in the faces and labels arrays.

**Training with the LBPH Model**We used OpenCV's LBPH (Local Binary Patterns Histogram) algorithm to train a face recognition model. LBPH operates by analyzing the local patterns within the grayscale pixel intensities of facial images, generating histograms that represent the unique characteristics of each face. These histograms are then compared during recognition to determine the closest match.

During training, the cropped grayscale face images and their corresponding labels (representing individuals) were passed to the LBPH model. The model processed the dataset and created a structured representation of each face. After successful training, the model parameters and data were saved to a file named face\_recognizer.yml. This file allows us to reuse the model without retraining, ensuring efficiency in subsequent testing and applications.

**Testing Process**

To evaluate the trained model, we tested it using a photograph not included in the LFW dataset. The following steps were taken during the testing phase:

**Preparing the Test Image**

The test image was resized to 250x250 pixels and converted to grayscale.

**Face Detection**

Face detection was performed on the test image, and the detected faces were passed to the model’s recognition function.

**Results**

The recognized person’s name and the model’s confidence level were displayed. Detected faces were highlighted with rectangles, and the names and confidence levels were overlaid on the test image.

**Evaluation of Results**

Unfortunately, the face recognition process using the LBPH algorithm did not achieve the expected success. During testing, we observed the following issues:

* **Incorrect Matches:** The model failed to match test individuals correctly and, in some cases, paired them with entirely different people.
* **Low Confidence Levels:** Prediction confidence levels were often very low, raising doubts about the model’s accuracy.

**New Direction: FaceNet**

With the limitations of LBPH in our face recognition project, we transitioned to a more advanced approach using the **FaceNet** model. FaceNet is known for its ability to learn discriminative features of faces by using deep convolutional neural networks (CNN). Unlike LBPH, which relies on local texture patterns, FaceNet generates embeddings for faces, which are high-dimensional vectors that represent the unique features of a face. These embeddings can then be compared using distance metrics, such as Euclidean distance, to determine the similarity between faces.

**Data Preparation for FaceNet**

In the new phase of the project, we used the **LFW (Labeled Faces in the Wild)** dataset again, as it is widely used in face recognition tasks. However, to achieve better results specific to our use case, we decided to include a **custom dataset** as well. This custom dataset contained data from specific users and conditions, ensuring that the model would perform better on the faces it was intended to recognize in the real world. The data preparation process involved face detection using the **MTCNN** model, followed by resizing the images to be compatible with the FaceNet model.

**FaceNet Training**

We employed the **Inception ResnetV1** model, which is a pre-trained model from the FaceNet architecture. This model has been trained on the **VGGFace2 dataset**, which is highly relevant to face recognition tasks. By leveraging this pre-trained model, we significantly reduced the amount of training data needed, which in turn saved considerable time and computational resources. Additionally, the **custom dataset** helped fine-tune the model to improve its performance on specific individuals.

**Embedding Extraction**

During the training phase, we extracted **face embeddings** from each detected face using the Inception ResnetV1 model. These embeddings represent each face as a vector in a high-dimensional space, where similar faces are located closer together. We saved these embeddings along with their corresponding labels (representing each individual), ensuring that we could use them in future testing and recognition phases.

**Testing with FaceNet**

After training the model, we tested it using real-time images. This **live testing** phase involved applying the trained FaceNet model to an unseen test image. The process was as follows:

1. **Face Detection:** We detected faces in the image using the **MTCNN** detector.
2. **Embedding Extraction:** For each detected face, we extracted embeddings using the FaceNet model.
3. **Face Matching:** These embeddings were compared against the stored embeddings from the training phase. Using Euclidean distance, we identified the closest match from the stored faces.

The **live testing** phase was implemented using a **webcam** and involved real-time face detection and recognition. The **FaceNet model** performed exceptionally well during this test, accurately identifying individuals under various conditions, and producing similarity scores. The following Python code snippet demonstrates the live testing process:The live testing demonstrated that the **FaceNet model** performed exceptionally well on real-world data, confirming its suitability for face recognition in various conditions.

**Evaluation of Results with FaceNet**

The results from using FaceNet were significantly improved compared to the LBPH model. The FaceNet model demonstrated:

* **Higher Accuracy:** The recognition rate improved, with the model correctly matching individuals in the test image more frequently.
* **Higher Confidence Levels:** The confidence levels during predictions were notably higher, suggesting a more accurate recognition process.
* **Better Generalization:** Unlike LBPH, which struggled with variation in lighting and pose, FaceNet performed better in recognizing faces under different conditions.

Additionally, the inclusion of the **custom dataset** and the application of **live testing** with these tailored images contributed significantly to the model’s ability to generalize to specific scenarios, further boosting its recognition performance.

While FaceNet provided much better results than LBPH, some challenges remained, particularly with faces in extreme poses or poor lighting conditions. However, with further tuning and possibly adding more training data, we were confident that FaceNet could be fine-tuned to reach even higher performance.

**Conclusion**

The decision to switch to **FaceNet** marked a significant improvement in the project’s face recognition capabilities. The model’s ability to generate face embeddings and compare them for recognition proved to be more robust than the traditional LBPH method. The **custom dataset** and the **live testing** phase have further solidified FaceNet’s effectiveness. As we continue to refine the FaceNet model and explore additional data augmentation techniques, we are optimistic about achieving even higher recognition rates and confidence levels, bringing us closer to the goal of highly accurate and reliable face recognition.

**References**

1. Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7), 971-987.  
   This paper introduced the Local Binary Patterns Histogram (LBPH) method, a widely-used approach in texture classification and face recognition.
2. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). *Facenet: A unified embedding for face recognition and clustering.* In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 815-823.  
   This paper describes the FaceNet model, which uses deep convolutional neural networks for face recognition by generating face embeddings.
3. Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). *Joint face detection and alignment using multitask cascaded convolutional networks.* In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2980-2988.  
   This paper introduces MTCNN, an effective face detection method, crucial for face alignment and localization before recognition.
4. Zhou, X., & Chellappa, R. (2017). *Face recognition in unconstrained environments.* Springer.  
   This book provides comprehensive insights into face recognition technologies, including both traditional and deep learning methods like LBPH and FaceNet.
5. Budiman, A., Fabian, R. A. Y., Achmad, S., & Kurniawan, A. (2022). *Student Attendance with Face Recognition (LBPH or CNN) Literature Review*. In *Proceedings of the 7th International Conference on Computer Science and Computational Intelligence*.