Exploring Facial Recognition

Aakash Kamuju AI21BTECH11001
Naresh Bandaru AI21BTECH1106
Mukunda Reddy AI21BTECH11021
Avinash Malothu AI21BTECH11018
Sumeeth CH AI21BTECH11008

April 26, 2024

Abstract

This paper thoroughly investigates various methodologies employed in face recognition, focusing on four prominent categories: holistic methods, feature-based approaches, hybrid models, and deep learning methods. The study aims to offer a comprehensive understanding of each category's strengths, and weaknesses.

1 Introduction

Facial recognition is the process of identifying or verifying individuals based on their facial features. It has gained significant attention due to its wide range of applications, such as access control, surveillance, and personalized user experiences. However, achieving accurate and reliable facial recognition systems re- mains challenging, particularly in unconstrained environments with variations in illumination, pose, expression, etc. extend this

2 Problem Statement

Traditional facial recognition methods often require extensive feature engineering to address challenges related to changes in orientation, pose, and expression of faces. This process can be complex and time-consuming, leading to suboptimal performance in real-world scenarios.

Deep learning techniques offer a promising solution to these challenges by automatically learning relevant features from raw data. However, the effectiveness of deep learning models heavily relies on the availability of large and diverse training datasets. By conducting a comprehensive comparative analysis, we aim to shed light on the strengths and weaknesses of each method.

3 Literature Review

We propose to implement and compare the following methods for facial recognition [Daniel and Li(2018)]:

3.1 Locality Preserving Projections (LPP)

LPP is a holistic method that aims to preserve the local structure of data while reducing its dimensionality. It finds a low-dimensional representation of the data that maximizes the local variance and minimizes the global variance. [X. He and Zhang(2005)]

3.2 Local Binary Patterns (LBP)

LBP is a feature-based method that encodes local texture patterns in an image. It computes a binary pattern for each pixel by comparing it with its neighbors and then histograms these patterns to represent the texture of the image. [T. Ahonen and Pietikainen(2006)]

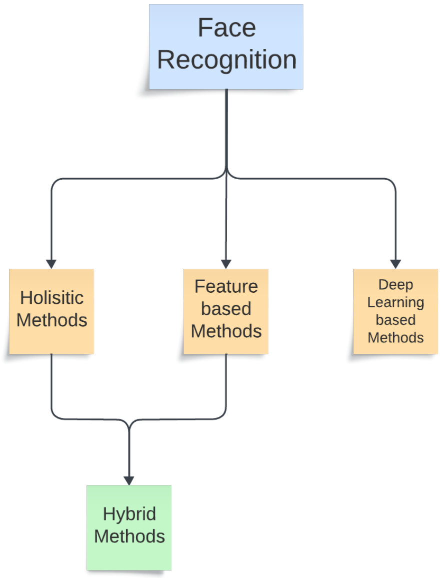


Figure 1: broad classification of face recognition methods

3.3 Gabor Feature based Classification using the Enhanced Fisher Linear Discriminant Model

This method utilizes Gabor filters to extract facial features followed by the enhanced Fisher linear discriminant model for classification. Gabor filters are tuned to capture texture information at different scales and orientations, and the enhanced Fisher linear discriminant model provides efficient feature extraction and classification. [Liu and Wechsler(2002)]

3.4 Deep Learning (Transfer Learning method)

We will also explore deep learning techniques for facial recognition using transfer learning. Transfer learning allows us to leverage pre-trained models on large datasets and fine-tune them on our facial recognition task. [Simonyan and Zisserman(2015)]

4 Experimental Plan

We intend to execute experiments utilizing established facial recognition datasets to assess the efficacy of various methods. Performance evaluation will encompass accuracy, computational efficiency, and resilience to variations in illumination, pose, and expression. Our dataset selection will prioritize diversity, incorporating variances in orientation, illumination, pose, and expression to ensure comprehensive and robust evaluation of the methods under consideration.

5 Methodology

We propose to implement and compare the following methods for facial recognition.

5.1 Locality Preserving Projections (LPP)

Here in LPP, we first perform PCA projection to reduce the dimensionality of the image set: $X_{PCA} = XW_{PCA}$, where X is the original data matrix, W_{PCA} is the PCA transformation matrix, and X_{PCA} contains the transformed images and then create a graph G with n nodes representing the images. Connect nodes i and j if x_i and x_j are among each other's k nearest neighborsand assign weights S_{ij} based on the Gaussian kernel: $S_{ij} = e^{-\frac{||x_i - x_j||^2}{t}}$ if nodes i and j are connected, otherwise $S_{ij} = 0$.

Compute the Laplacian matrix $L = D - S$, where D is a diagonal matrix with entries as column sums of S. Solve the generalized eigenvalue problem: $XLX^T w = \lambda XDXTw$, where X is the matrix containing the PCA-transformed images. Order the eigenvectors and eigenvalues by their corresponding eigenvalues (λ).

Create the transformation matrix $W = [W_{PCA}, W_{LPP}]$, where W_{LPP} contains the eigenvectors corresponding to the smallest eigenvalues of the Laplacian matrix. Perform linear mapping: $y = W^T x$, where y represents the Laplacianfaces.

We first implemented the Laplacianfaces algorithm, which is a dimensionality reduction technique

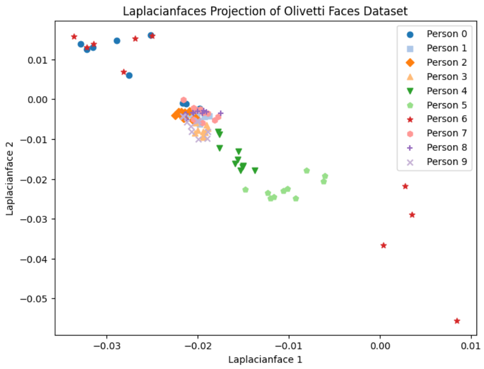


Figure 2: LPP output

used particularly for face recognition tasks.The algorithm begins by computing pairwise distances between the images in the dataset and constructs a similarity matrix based on the distances, where closer images are assigned higher similarity values. This is achieved by considering each image's k nearest neighbors and applying a Gaussian kernel to compute the similarities.

Next, the algorithm constructs a Laplacian matrix by subtracting the similarity matrix from a diagonal matrix, where the diagonal entries are the sums of the similarity matrix's columns. Eigenmap is then applied to this Laplacian matrix, solving a generalized eigenvalue problem to obtain eigenvectors and eigenvalues. These eigenvectors represent the "Laplacianfaces" which capture the local structure of the data. The final output of the algorithm on the dataset Olivetti faces is shown in Figure 7.

5.2 Local Binary Patterns (LBP)

5.2.1 Introduction

Local Binary Pattern (LBP) is a texture-based approach. Above paper explores the application of LBP in facial recognition, comparing its performance against traditional raw pixel data methods.

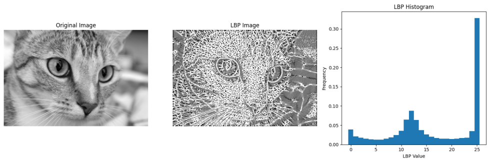


Figure 3: image, LBP image, histogram

5.2.2 LBP Computation

Local Binary Patterns (LBP) is a powerful, simple, and efficient texture operator that labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. It has gained popularity in facial recognition due to its robustness to monotonic gray-scale changes and computational simplicity.
Given a pixel in the image, the LBP value is computed by comparing its intensity with that of its neighbors. Consider a 3x3 neighborhood of pixels. The center pixel's intensity is compared with its eight neighbors. If a neighbor's intensity is greater than or equal to the center pixel, it is marked as 1; otherwise, it is marked as 0. This binary result forms the LBP code for the center pixel. This process is repeated for every pixel in the image to obtain the LBP image. The histogram of these LBP codes then serves as a feature vector for the image.

5.2.3 Example

An example of applying LBP to a grayscale image and its corresponding histogram is shown in Figure 3.

5.2.4 Results and Discussion

The study utilized the LFW (Labeled Faces in the Wild) dataset to evaluate the performance of facial recognition using both raw pixel intensities and LBP features. Support Vector Machines (SVM) were employed as the classifier in both scenarios. The classification reports, and accuracy metrics reveal that the LBP features method outperforms the raw pixel data approach. Specifically, the LBP method achieved an accuracy of 91.74%, compared to 90.00% with raw images, as shown in Table **??**.

The improvement in accuracy with LBP features can be attributed to the method's ability to capture essential texture information that is more discriminative for facial recognition than raw pixel intensities. The texture features encoded by LBP are particularly effective in distinguishing between different faces due to the unique patterns of edges, spots, and other facial attributes that they capture.

	precision	**recall**	**f1-score**
False	0.74	0.98	0.84
True	0.97	0.66	0.79
accuracy			0.82
macro avg	0.86	0.82	0.82
weighted avg	0.86	0.82	0.82

Table 1: Classification report

5.3 Gabor Feature-based Classification using the Enhanced Fisher Linear Discriminant Model

The proposed GFC method leverages the Gabor wavelet representation to construct an augmented feature vector, which is then processed using EFM for dimensionality reduction while enhancing discrimination power. By concatenating features from different Gabor kernels, the augmented feature vector captures rich facial information. Subsequently, EFM selects the most discriminative features, leading to a compact representation suitable for classification tasks. The feasibility of the GFC method is demonstrated through experiments on the FERET database, showcasing superior performance compared to existing methods

5.3.1 Gabor Filters

The first step in the Gabor Fisher Classifier is obtaining the Gabor feature vector. This is done through convolution with a family of Gabor kernels with different spaces and orientations. We spent some time understanding what Gabor kernels are and looking at their mathematical equations.

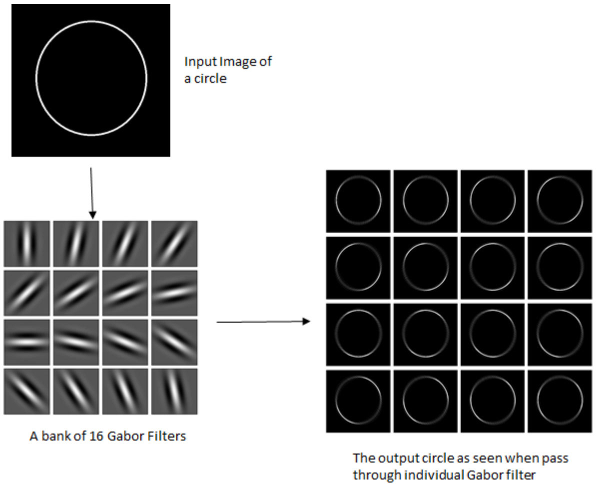

Input Image of a circle
A bank of 16 Gabor Filters
The output circle as seen when pass through individual Gabor filter

Figure 4: Gabor Filter

The Gabor wavelets (kernels, filters) can be defined as follows

$$\psi_{\mu,\nu}(z) = \frac{\|\mathbf{k}\mu,\nu\|^2}{\sigma^2} e^{-\frac{\|\mathbf{k}_{\mu,\nu}\|^2\|z\|^2}{2\sigma^2}} \left[e^{i\mathbf{k}_{\mu,\nu}\cdot z} - e^{-\frac{\sigma^2}{2}} \right] \tag{1}$$

where μ and ν define the orientation and scale of the Gabor kernels, $z = (x, y)$, and $\| \cdot \|$ denotes the norm operator, and the wave vector $\mathbf{k}_{\mu,\nu}$ is defined as follows:

$$\mathbf{k}\mu,\nu = k\nu e^{i\phi_\mu} \tag{2}$$

where $k_\nu = k_{max}/f^\nu$ and $\phi_\mu = \pi\mu/8$. k_{max} is the maximum frequency, and f is the spacing factor between kernels in the frequency domain

5.3.2 Dimensionality Reduction

Following the initial dimensionality reduction using PCA, the enhanced Fischer linear discriminant (EFM) is subsequently applied to the augmented Gabor feature vector. EFM offers a distinct advantage in effectively capturing discriminative information within the feature space, resulting in a low-dimensional representation with enhanced discrimination power. This approach not only streamlines the feature extraction process but also ensures that essential discriminatory features are preserved, thereby facilitating more accurate classification. It's worth noting that training and testing were conducted using the Labeled Faces in the Wild (LFW) dataset, a widely recognized benchmark for face recognition tasks.

5.3.3 Classifier

In the classification phase, Support Vector Machines (SVM) are employed with a linear kernel to efficiently classify the final feature vectors. By utilizing a linear kernel, SVM can effectively separate data points in the reduced feature space, enabling robust and accurate classification even in complex datasets.

5.3.4 Testing and Results

The model evaluation was conducted on a dataset comprising pairs of images, where the task involved determining whether the images within each pair belong to the same individual. The model classifies each image pair and compares the predicted classes to verify if they correspond to the same subject. The classification report detailing the model's performance metrics for this task is provided below.

	precision	**recall**	**f1-score**
False	0.74	0.98	0.84
True	0.97	0.66	0.79
accuracy			0.82
macro avg	0.86	0.82	0.82
weighted avg	0.86	0.82	0.82

Table 2: Classification report

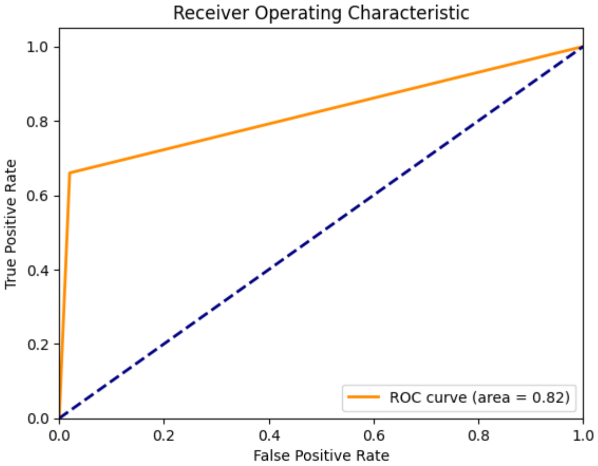


Figure 5: ROC

5.4 Deep Learning Method

We present our approach to facial expression recognition using the Yale dataset, employing a Siamese Network for facial expression recognition using transfer learning with a pre-trained VGG16 model trained with triplet loss. The Yale dataset is chosen for its diversity in facial expressions, providing a robust platform for training and testing our model.

5.4.1 Yale Dataset

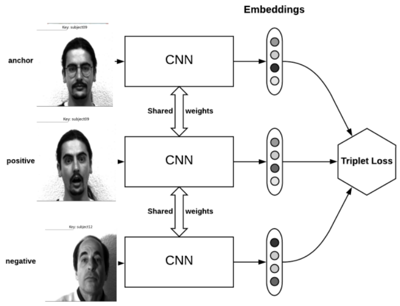
The Yale dataset is a widely used benchmark dataset in the field of facial recognition. It contains high-resolution images of individuals under various lighting conditions, poses, and facial expressions. This diversity makes it suitable for training models robust to real-world scenarios.

5.4.2 Network Architecture

Traditional neural network classifiers for face recognition can encounter challenges when dealing with datasets containing numerous individuals. Each person in the dataset becomes a class, leading to a high number of classes, which can be inefficient and demanding in terms of computational resources. Moreover, training such classifiers on a large dataset requires substantial time and effort. This is also the case with classical techniques.

In contrast, Siamese networks offer an elegant solution to these challenges. Instead of directly classifying images, Siamese networks focus on learning similarities between pairs of images. They achieve this by comparing the features extracted from two input images and determining how similar or dissimilar they are.

A Siamese network is a type of neural network architecture that consists of two identical subnetworks, each taking a different input. These subnetworks learn to extract features from the input data and are then compared at a feature level to determine similarity or dissimilarity.



[t]

Figure 6: Network Architecture

[Yaniv Taigman(2014)]

Transfer learning represents a nuanced approach in machine learning, leveraging insights acquired from training one model to enrich the performance of another model tackling similar tasks. This methodology capitalizes on pre-trained models and their accumulated knowledge to streamline the training process of subsequent models while augmenting their ability to generalize across a spectrum of tasks. In our approach, we employed the VGG-16 architecture as the foundation for our convolutional neural network (CNN). Enhancing this architecture, we appended an additional convolutional layer at the backend and introduced two linear layers at the front end. The final layer of the network outputs embeddings of length 128, encapsulating the essential features extracted from the input data. This fusion of pretrained knowledge with task-specific adjustments enables our model to swiftly adapt to new tasks while fostering robust generalization capabilities across diverse domains.

5.4.3 Triplet Loss

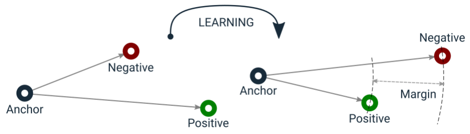
Triplet loss is a loss function commonly used in Siamese network training. It works by comparing three samples: an anchor sample, a positive sample (similar to the anchor), and a negative sample (dissimilar to the anchor). The objective is to minimize the distance between the anchor and the positive sample while maximizing the distance between the anchor and the negative sample. This encourages the model to learn embeddings where similar samples are closer together and dissimilar samples are farther apart. [Florian Schroff(2015)]

$$\|f(x_{a_i}) - f(x_{p_i})\|_2^2 + \alpha < \|f(x_{a_i}) - f(x_{n_i})\|_2^2,$$

$$\forall (f(x_{a_i}), f(x_{p_i}), f(x_{n_i})) \in T$$

where α represents the enforced margin between positive and negative pairs, and T denotes the set of all possible triplets in the training set with cardinality N. The loss function to be minimized is defined as:

$$L = \sum_{i=1}^{N} \left(\|f(x_{a_i}) - f(x_{p_i})\|_2^2 - \|f(x_{a_i}) - f(x_{n_i})\|_2^2 + \alpha \right)$$



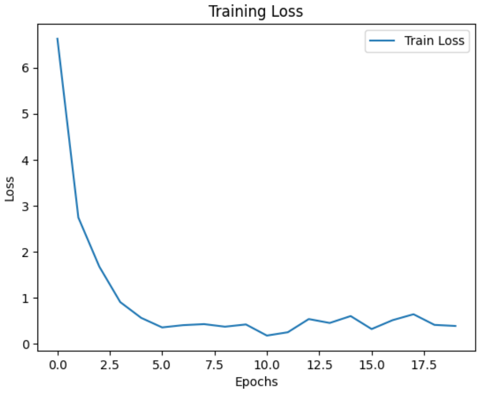
[t]

Figure 7: triplet loss

5.4.4 Sample Selection using Semi-Hard Samples

In our implementation, we adopted the strategy of semi-hard sample mining during training. Semi-hard samples are those for which the negative sample is farther from the anchor than the positive sample but still contributes to the loss. By selecting semi-hard samples, we strike a balance between encouraging meaningful updates to the model and preventing it from being overwhelmed by excessively difficult examples.

our training regimen unfolds with the generation of embeddings after each epoch using the updated model. Subsequently, we curate semi-hard triplets from these embeddings, facilitating the fine-tuning of the network. The training process continues iteratively until the loss converges to a stable state. The ensuing results showcase a discernible pattern wherein the loss stabilizes after a handful of epochs, signifying the convergence of the training process.



[t]

Figure 8: Training Loss

After approximately 18 epochs, employing a batch size of 32 and comprising a total of 50 batches, we observe the error curve reaching a state of stabilization, aligning with our desired outcome. This signifies the convergence of our training process, wherein the network attains a consistent level of performance, indicative of its optimized state.

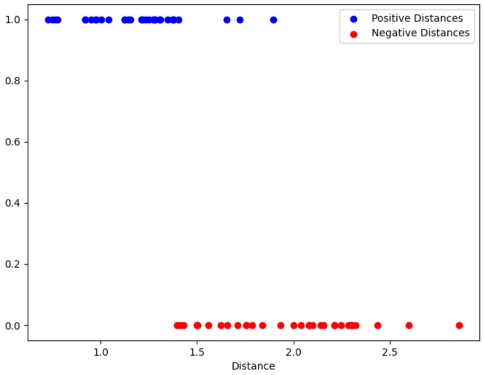
5.4.5 Validation

Embedding Extraction: Once trained, the Siamese network is used to extract embeddings for each face image in the dataset. These embeddings represent the unique features of each face in a lower-dimensional space.

Distance Measurement: The distance between embeddings is calculated using a chosen metric, such as Euclidean distance or cosine similarity. Smaller distances indicate greater similarity between faces, while larger distances signify dissimilarity.

Threshold Selection for Classification: Now, to use these embeddings for face recognition, we need a threshold to classify whether two faces belong to the same person or not. This threshold is crucial for determining the boundary between positive and negative samples.

Threshold Selection: The threshold is typically selected based on an evaluation metric, such as accuracy, precision, or F1-score, using a validation dataset. The goal is to find the threshold that maximizes the chosen metric.



[t]

Figure 9: distance between the similar images and dissimilar images

5.4.6 Testing

The classification report here evaluates the performance of a model on a binary classification task, where class 0 represents negative samples(images of different people) and class 1 represents positive samples(images of same person). The precision, recall, and F1-score metrics provide insights into how well the model performs for each class and overall.

	precision	**recall**	**f1-score**
0	0.85	0.88	0.86
1	0.87	0.84	0.86
accuracy			0.86
macro avg	0.86	0.86	0.86
weighted avg	0.86	0.86	0.86

Table 3: Classification report

Our model demonstrates proficiency in both class 0 and class 1, evident in the commendable precision and recall scores. This adeptness extends seamlessly to the testing data, underscoring the model's robust generalization capabilities. Moreover, as the dataset size expands, the efficacy of our model becomes increasingly apparent, yielding markedly improved results.

6 Conclusion

Gabor feature-based classification, coupled with the Enhanced Fisher Linear Discriminant Model, is effective in addressing data challenges related to illumination and facial expression changes. However, we hypothesize that the presence of pose variations within the Labeled Faces in the Wild (LFW) dataset may have contributed to the observed accuracy limitation, which stabilized at 82%. Further investigation into the impact of pose variations on classification accuracy could provide valuable insights for future improvements.

References

[Daniel and Li(2018)] Daniel and Li. Face recognition: From traditional to deep learning methods. 2018. 1

[Florian Schroff(2015)] James Philbin Florian Schroff, Dmitry Kalenichenko. Facenet: A unified embedding for face recognition and clustering. 2015. 6

[Liu and Wechsler(2002)] C. Liu and H. Wechsler. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. 2002. 2

[Simonyan and Zisserman(2015)] K. Simonyan and A. Zisserman. “very deep convolutional networks for large-scale image recognition. 2015. 2

[T. Ahonen and Pietikainen(2006)] A. Hadid T. Ahonen and M. Pietikainen. Face description with local binary patterns: Application to face recognition. 2006. 1

[X. He and Zhang(2005)] Y. Hu P. Niyogi X. He, S. Yan and H.-J. Zhang. Face recognition using laplacian faces. 2005. 1

[Yaniv Taigman(2014)] Marc'Aurelio Ranzato Lior Wolf Yaniv Taigman, Ming Yang. Deepface: Closing the gap to human-level performance in face verification. *CVPR*, 2014. 6