Article title: Overview of Face Recognition Methodologies: A Literature Review

Authors: Akshay Menon[1]
Affiliations: student at presidency university, bengaluru[1]
Orcid ids: 0009-0001-2656-1960[1]
Contact e-mail: akshaymjry@gmail.com

DOI: 10.14293/PR2199.000346.v1
Preprint first posted online: 08 September 2023
Keywords: Face Recognition, Computer Vision, Deep Learning, Eigenfaces, Local Binary Patterns, Sparse Representation, Robust Recognition, Evaluation Methods, Image Processing, Feature-based Methods

Overview of Face Recognition Methodologies: A Literature Review

Akshay Menon
Presidency University, Bengaluru
Email: akshay.20201csd0033@presidencyuniversity.in

***Abstract*—Computer vision and pattern recognition technologies have made a significant advancement in the recent past. A key application of this is face recognition. Its applications resonate in many fields. Various methodologies have been developed for this, and it is essential to compare, contrast and review them on the basis of various factors and aspects to find the optimal methodology. The journey of face recognition technology has progressed rapidly from traditional methods to deep learning applications. Yet, it is necessary to survey all these methods and critically and comprehensively evaluate them, as a majority of them are still limited to variation constraints such as lighting, expressions and orientation and other practical complexities. The objective of this paper is to examine, evaluate and illustrate the strengths and limitations of a range of papers by delving into various distinct yet comparable specifics of each methodology of implementing face recognition systems. The motive is to also culminate the essence of these methodologies to facilitate an insightful perspective and overview of the methodologies in a concise manner.**

I. Introduction

Face recognition is a fundamental example of utilizing computer vision technologies with applications ranging from security, surveillance, legal, entertainment and other diverse yet integral parts of daily activities. The increasing dependence on the potential of face recognition has led to a lot of research in the field, resulting in significant developments and advanced systems to yield this potential to the fullest. Different methods and algorithms have been developed, each having its own unique way of extracting features from images to obtain patterns from them and implement recognition. To broadly categorize them, face recognition methodologies are generally of four major types. Firstly, there are **holistic methods** that focus on treating the face as a single entity and obtaining the overall characteristics from it. Followed by this, there are also **feature-based methods** that involve the extraction and analysis of particular facial features like the eyes, mouth, or nose. Thirdly, **template-based methods** create a representation of the face using the distinct features based on edge patterns, etc. Lastly, **hybrid methods** integrate multiple methodologies to build a more robust recognition system and harness the strengths from each of its derivatives. This paper explores methodologies from all these categories and also discusses a variety of methods of optimizing various aspects such as feature representation and dimensionality reduction.

II. Methodologies

A. Local Binary Patterns

The face is a complex surface-structure with an extremely high degree of distinctness across different individuals. A computer cannot explicitly identify a face as on its own. Developing a face recognition model involves training the system with some input set of features derived from the geometrical appearance patterns of the face. This calls for the need to maximize relevant features to build an efficient and optimized model. Every image consists of pixels which have different intensities, and local binary patterns are the feature of interest in this methodology. These patterns capture and encode the edges on a face in an intuitive manner. The pixels of each image in a face can be grouped into sequential windows of 3×3 pixel-blocks. By comparing its intensity with the central pixel of each block, the surrounding pixels can be normalized to either 0 (if its intensity is lesser than central pixel) or 1 (if intensity is greater than central pixel). This converts each pixel block into a series of binary values with corresponding decimal values. A histogram of these values helps visualize the "information about the distribution of the local micro-patterns, such as edges, spots and flat areas, over the whole image". [1] The transitions between binary values signify the presence of an edge. Through this, the otherwise high-dimensional face image is now encoded into a low-dimensional space by utilizing relative intensity values, also known as illumination invariant descriptors. However, this doesn't solve the problem of facial expression detection because expressions are an action that occurs in a span of time. Hence, the 3×3 blocks are extended into 3×3×3 'cuboids' (analogy to quantify space-time) which ensures the third dimension of time is taken into account. Introducing this dimension increases the number of possible byte values from 2^8, to a significantly large 2^{26} values. The solution to simplifying this is by only considering the orthogonal plane pairs (XY, XZ and YZ) where each plane encodes an edge. Now, the possible byte values have been reduced to a far more computationally efficient, $3{\times}2^8$ values, while precisely encoding the facial movements and capturing expressions.

Above mentioned is a simplified abstract of the working of local binary patterns-based face recognition. This methodology has been experimentally studied extensively in the research paper by T. Ahonen *et al.* [1]. In their experiment, local binary pattern-based texture analysis was conducted on the FERET dataset [2] by measuring intensity changes in 3×3 pixel blocks using the LBPu2 operator with varying window sizes to spatially represent the face image. The dissimilarity measures used were histogram intersection, log-likelihood and chi-square statistic (χ^2). The two key statistics produced in this study were mean recognition rate and probabilities to compare LBP performance against different algorithms including Principal Component Analysis (PCA) [3], Linear Discriminant Analysis (LDA) [4], Bayesian Intrapersonal Classifier (BIC) [5] and Elastic Bunch Graph Matching (EBGM) [6].

The findings of this paper are significant in proving the efficacy of local binary patterns-based face recognition. As per the study, LBP is proved to be robust with a recognition rate of 79% against 65%, 37% and 42% for PCA, BIC and EBGM. Histogram intersection and χ^2 dissimilarity measures are preferred for smaller window sizes, while log-likelihood is more suitable for larger window sizes. Local binary patterns are also an ideal method of recognition even in cases of high degrees of variation in parameters like lighting, expressions and aging based on results of tests under variations on the Olivetti Research Laboratory (ORL) face database [7]. Another key advantage of the proposed approach highlighted in the paper is the "simplicity of the proposed method that allows for very fast feature extraction."

B. Eigenfaces

Principal component analysis (PCA) is a fundamental technique to implement dimensionality reduction on high-dimensional datasets. The structural complexity of the face roots from its intricacies and distinctness (as previously mentioned) also indicates the high dimensionality of facial data. One approach to simplify this would be to reduce the dependency on the detailed geometry of the face. A way to achieve this is to remove redundant appearance features and only focus on the distinct ones. PCA can be implemented through a simple yet intuitive manner. To begin, a set of images are first converted into greyscale which eliminates color information but preserves the structural information while reducing the complexity. The pixel intensities in a greyscale image are represented as matrices. Every image in a dataset will have its own unique matrix. The **mean face** is calculated, that produces a single image (matrix) of the average pixel intensity across all images in the training dataset. The mean face is subtracted from each original face image which produces a new set of images (difference images) which only contains the variations. Now, the dimensionality and complexity of each image in the entire training set has been largely reduced to facilitate computational efficiency. Using the new set of difference images, the covariance matrix is calculated. This is used to to compute eigenvalues and eigenvectors. These eigenvectors are referred to as **eigenfaces**. The most significant eigenfaces are selected to capture important variations. Each face image is represented as a linear combination of its eigenfaces. Using PCA, these images are projected into a new face-space where the original images are reconstructed only using the selected (weighted) eigenfaces. The face space "does not necessarily correspond to isolated features such as eyes, ears, and noses." [8]. This establishes a model for face image classification. To classify new test images, a suitable distance metric (like euclidean distance) between the image and its projection in the subspace can be used.

The paper of interest is an experimental study of the eigenfaces-based methodology by M. A. Turk *et al.* [8]. The authors have mentioned that this idea roots from a technique developed by Sirovich and Kirby [9] "for efficiently representing pictures of faces using principal component analysis" where "a collection of face images can be approximately reconstructed by storing a small collection of weights for each face and a small set of standard pictures." Their study involves a set of two experiments. The first is based on examining the effects of varying illumination, image size, and head orientation. The second experiment aims at measuring the effects of varying acceptance threshold (θ_c) for the same parameters as the first experiment, also to measure the trade-offs. The database used for training consists of a set of 2,500 face images of sixteen subjects at different head orientations, scales, and illumination. A six-level multiscale approach is implemented using a Gaussian pyramid for pixel resolutions ranging from 512×512 to 16×16.

The findings of the study helps to understand the relative robustness and stability of this approach. For infinite θ_c, lighting variation results in 96% accuracy, orientation variation results in 85% accuracy, and size variation results in 64% accuracy (on average). Reducing θ_c to achieve 100% recognition accuracy resulted in unknown rates of 19% while varying lighting, 39% for orientation, and 60% for size. Higher θ_c resulted in higher degree of error. Eigenfaces prove high accuracy at arbitrary unknown rates with 100%, 94%, and 74%. This reflects that lighting changes result in few errors, but size changes significantly affect performance (hence needs multiscale comparison). This shows that the eigenface approach performs exceptionally well in handling variations. On the other hand, the paper highlights some potential weaknesses of this methodology too. The study indicates that noisy images proved to negatively affect performance as the 'auto-associative' memory of the system gets diluted. Another drawback is the fact that the system inherently tends to store the patterns of images that are classified as 'unknown' in the pattern space as well, which may seem counterproductive.

C. Relaxed Local Ternary Patterns

As previously seen, local binary patterns are only based on the distribution of magnitude of differences in intensity. The local ternary patterns-based methodology (LTP) is similar, but the pixel intensity variations are stored in a separate threshold state that takes both sign and magnitude into account and results in a positive and a negative LBP histogram. As mentioned in the paper by J. Ren *et al.* [10], LTP is "less sensitive to noise" and "the dimensionality of LTP histogram is very large" resulting in "a histogram of 3^8= 6561 bins". The key drawback of LBP and LTP is redundancy in Relaxed local ternary patterns are a modification and an extension of the ternary pattern methodology through the introduction of a third parameter known as the **uncertain state**. This uncertain state is used to hold off small pixel intensity differences to counter the drastic bit-value changes reflected in local binary and ternary patterns. Hence, the 'relaxed' approach provides a stronger grip against noise by ensuring a smoother distribution of pixel intensities. The uncertain state holds a value of either -1, 0, or 1. This results in higher dimensionality which raises the question of how it can be optimized for computational efficiency. To solve this, the paper of interest proposes the idea of converting the trinary value back to binary by "encoding State X equally into two strong states, i.e. State 0 and 1 with equal probability." Note that Here, "State X" refers to the trinary uncertain state.

In the paper by J. Ren *et al.* [10], the relaxed local ternary pattern approach (RLTP) has been experimentally studied over three datasets, CMU-PIE (Pose and Illumination) dataset [11], extended Yale B dataset [12] [13] and the O2FN mobile face dataset [14]. The images are preprocessed through gamma correction, difference of Gaussians filtering, and contrast equalization. The study has been conducted under varying noise (p) settings. The measurements are based on comparing recognition rates against optimal threshold using Chi-square distances and comparison RLTP results against LBP and LTP.

The findings of this paper indicate RLTP produces a recognition rate of 98.40% against 96.60% for LBP and 97.40% for LTP. This proves its higher accuracy, tolerance to noise, and overall improvement in performance over its counterparts. They have mentioned that the optimal threshold for LTP ranges from 6 to 10 for different databases, but suggest that the optimal threshold be selected as 2. Hence, the result of this study reinforces the robustness of this improved technique to noise.

D. Deep Learning (Convolutional Neural Networks)

Deep learning is a significant subsidiary of machine learning under the umbrella of artificial intelligence. By harnessing the power of neural networks, extensible and dynamic learning models can be developed to perform tasks like face recognition. The focus in this section will be on the implementation of face recognition using a convolutional neural networks (CNN) based approach. Before delving into CNN, it's important to revisit the fundamental principles of deep learning and neural networks and their applications in computer vision technology. Generally in traditional learning, we focus on combinations of independent variables (features) that contribute to a certain outcome parameter that we want to predict or obtain (using methods like regression). This means utilizing more relevant features ideally helps develop a more accurate model, but its output generation is limited to the features that have been explicitly defined during training. On the other hand, the objective of deep learning is to develop a model that can 'learn' from the inherent correlations between explicitly defined features, which may seem 'hidden' to us, but impact the outcomes as an intermediate. This means there would be a greater number of 'nodes' that form a deep neural network of related parameters. The arrangements of these nodes form layers that lead to more abstract and complex feature representations.

The concept of **convolution** can be understood through its applications in computer vision. Kernel convolution is a technique used to apply filters and blurs on images, where the filter is just small 'grid' of different combinations of numbers based on the effect we intend to make on an image. The filter is passed through an image, which transforms it accordingly. To better understand, consider a 3×3 grid kernel. This kernel window slides along corresponding 3×3 regions of the image. In each instance, the corresponding kernel and image pixel values are multiplied. This demonstrates convolution, which by definition involves a mathematical operation (generally multiplication) that is done between values of the filter and the corresponding values of the input grid.

From the perspective of applying neural networks in computer vision, a common presumption would be that each pixel of an image can be assigned to an input node in the neural network. But knowing images generally comprise millions of pixels, it is not computationally feasible to execute this. This means that an alternative way of feature representation must be used. This is where convolution plays its role. Convolutional neural networks essentially involve replacing each node with a kernel convolution process. Practically, we start with an image, and slide along the image with a fixed window size like 3×3 pixels, capturing patterns and generating a feature map that comprises edge/corner patterns and other textural information. Each iteration involves a kernel convolution process that generates a new feature map, and this is how the convolutional neural network gets more complex and accurate in identifying textural patterns. There are three types of layers in CNN. Firstly, the convolutional layer that detects patterns. Then, the pooling layer that simplifies the patterns detected by the convolutional layers. Lastly, the fully connected layer that makes sense out of these simplified patterns and interprets the learned representation, hence, producing the output of face recognition.

The paper of interest in this context is a comprehensive and detailed experimental study by P. S. Prasad *et al.* [15] This study proposes two approaches. First is the VGG-16 Face Network [16] (deep convolutional network) trained on 2.6 million face images, 2522 people with 16 convolution layers, 3 fully connected layers, 5 pooling layers and a final Softmax activation layer to compute class probabilities. The paper also cites that VGG is a 'computationally costly design'. [17] The other approach is the **lightened CNN** design with lower-computational complexity, utilizing Max-Feature Map (MFM) activation and also Softmax linear activation layer at the output. It has been implemented through two models: the AlexNet model (3962 K parameters, 4 convolutional layers, 4 max-pooling layers, and 2 fully-connected layers) and the Network in Network-inspired model (3245 K parameters, 4 convolutional layers, 5 max-pooling layers, 2 fully-connected layers). The evaluation involves testing the robustness of lightened CNN and VGG-16 recognition on the AR face database [18] under varying conditions of occlusion (wearing scarf, sunglasses) in different sessions. Note that lightened CNN classification has been evaluated on two classification layers, FC6 and FC7.

The classification results obtained from this study (from 'Table 1' [15]) have been abridged to their mean values for simpler presentation in this review. Firstly, the average rates of recognition for lightened CNN were found to be 85.88% for FC7 Scarf sessions, 28.68% for FC7 sunglasses sessions, 88.79% for FC6 Scarf, 32.27% for FC6 sunglasses. Hence, FC6-lightened CNN produced better results. On the other hand, the average rates of recognition for VGG were: 11.31% for Scarf sessions and 6.25% for sunglasses. This shows that lightened CNN provides far superior recognition rates over VGG in cases of the presence of occlusion. Another key takeaway from the conclusions of this study is the fact that this deep learning approach also provides "more robustness to misalignment of facial images."

E. Fisherfaces

We previously saw the application of eigenfaces implemented with principal component analysis (PCA) for dimensionality reduction. Other dimensionality techniques are also commonly used, coupled with other methodologies. One such example is **fisherfaces** implemented by combining linear discriminant analysis (LDA) and PCA, introduced by Belheumeur [19]. LDA is a dimensionality reduction technique to reduce and capture only essential features from a dataset. Like in the case of PCA, even LDA maximizes separation between different classes after projection into a feature space. This feature space is then projected onto a smaller subspace while maintaining class-discriminating information, which is not a property of a PCA. PCA is not as efficient in the separation between classes. [19] [3], hence reinforcing the advantage that LDA exhibits over PCA. The computation of fisherfaces involves a series of steps. First, the **within-class** scatter matrix (S) is computed. This matrix captures how well the data is scattered within its class, represented as the mean of each category. This is calculated through the summation of covariance matrices (S_i) of n classes represented by $S_w = \sum_0^i S_n$. The next step involves computing the **between-class** scatter matrix that captures scattering across classes (S_b) based on differences in class means, given by

$S_b = \sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T$,

where

N_i is the number of samples in the i^{th} class,

μ_i is the mean of i_{th} class projected onto a feature subspace defined by $S_W^{-1} S_B$, and

μ is the overall mean.

This feature subspace comprises projection vectors obtained by finding eigenvectors that maximize the ratio of between-class and within-class scatter to maximize class discrimination. From this, the projection vectors that best represent the maximized class separation are identified. These vectors carry relevant features that represent the dataset. Test data is projected onto the feature subspace and classified using a suitable distance metric and classification technique.

The paper that discusses the study by M. Anggo *et al.* [20] delves into the experimental analysis of the application of Fisher's Linear Discriminant (based on Linear Discriminant Analysis) applied with PCA-based dimensionality reduction on images of people from the Papuan population [21]. The recognition algorithm has been developed using MATLAB 7.10, and Adobe Photoshop CS4 is used for image preprocessing. The evaluation method used in this study is Euclidean distances between feature vectors of test and train images for classification, and the percentage of correct recognitions over total test images to measure the model's accuracy.

The results of this study show that the model shows a 100% recognition rate when the testing image is the same as the training image and 93% accuracy was obtained when using images with varying expressions and positions. This methodology is robust against noise and blurring. It is mentioned that errors mainly arose due to variations in scaling and poses which can be improved using better image scaling and expanding the training set respectively.

F. Histogram of Oriented Gradients

Any image can be represented in its image function form $f(x, y)$. For this function, let's represent each image pixel's gradient as (f_x, f_y). The gradients essentially capture the changes in intensity across the entire image, hence they can be used to capture and visualize the edge patterns and its textural information. When visualizing this edge map, positive gradients (lower to higher intensity) are generally represented as a white edge and negative gradients form a black edge. The direction of the gradient vector at any pixel in an image is given by

$\arctan(f_y/f_x)$

and its magnitude is given by

$\sqrt{(f_x)^2 + (f_y)^2}$.

The significance of gradients to identify textural information, yet it is important to understand how it is practically applied. For example, we first take an image and consider a single pixel that is to be examined. The first objective is to calculate the horizontal (x) and vertical (y) gradients. This can be done by finding the change in intensity values around that central pixel and their respective directions. Given below is an example of this calculation:

	80	
50	P	140
	40	

Figure 1. Gradient Calculation Example

The diagram above shows all neighboring pixel intensities around a central pixel P. The horizontal and vertical gradients of the central pixel can be calculated by simply finding the respective intensity differences of the pixels around it. Here, the horizontal gradient (f_x) is $140 - 50 = 90$. The vertical gradient (f_y) would be $80 - 40 = 40$. Hence, the gradient feature vector for the central pixel P is [90 40]. The approximate magnitude for this feature vector is 98.5 units and its direction is 24°. This gradient vector itself is the **oriented gradient**, which lays the basis for the concept of histogram of oriented gradients (HOG). An image is divided into smaller n×n cells. For example, if 8×8 pixel cells are used, 64 gradient vectors are computed from each cell using which a histogram of frequency of gradient vectors into a smaller number of bins (that represent both magnitude and direction). Let's say the number of bins are 8, then the 64 values have been essentially 'condensed' to just 8 values through its histogram representation. This reduces the computational cost to train a machine learning model by using a low-dimensional feature representation of the image.

The paper of interest in this review is by O. Déniz *et al.* [22], where the use of histogram of gradients based descriptors has been studied extensively through experiments on the FERET [2], CMU Multi-PIE 2 [11], AR [18] and Yale [23] facial databases. The study involves two key experiments. The first experiment aims to evaluate the effect of face landmark localization error on recognition accuracy by comparing HOG features located from landmarks (distinct facial features) localized using Active Appearance Models (AAM) [24], which is bound to a degree of error), against HOG features from a regular grid. This experiment helps understand robustness to facial feature location by comparing performances of holistic verses HOG-based representation. The second experiment examines effect of regular grid HOG feature extraction at multiple scales using PCA and LDA for dimensionality reduction. Various occlusions and their impact on recognition performance have also been examined such as variations in lighting, aging, expression, illumination and the presence of accessories such as scarves, glasses.

The results obtained in this study reflect that the best recognition rates were measured on the FERET database with HOB-EBGM [25] approach with an accuracy of 95.5%, which reinforces the robustness of the HOG approach. The computational cost to extract HOG features was ideal when using larger patch sizes (>12×12) from the regular grid, but the best recognition performance was by using combinations of different patch sizes rather than a single ideal size.

G. Local Directional Patterns

Local directional patterns (LDP) are a variation of local binary patterns. The key difference is that LDP uses **edge response** values while LBP focuses on pixel intensities. LDP can be used as a local feature descriptor that encodes edge response values based on intensity differences and their directional information. By definition, edge response values are a measure of change in pixel intensity in a particular direction. For example, a high response value indicates the presence of a corner or edge. These are implemented by applying 'masks' to quantify variation in pixel intensities in particular different directions (typically eight) which can be used to encode the local textural information of an image.

To understand the application of local directional patterns in face recognition, we will analyze the paper by T. Jabid *et al.* [26] that is based on a study involving the use of a novel local feature descriptor to obtain LDP features through computation of edge response values. In this study, the evaluation has been conducted on the FERET image database [2] using the CSU Face Identification Evaluation System. Each image is normalized to 100x100 pixels, then processed in 10x10 blocks. The images are divided into four probe sets fb (expression variation), fc (illumination variation), $dupI$ (age variation) and $dupII$ (age variation). The study also focuses on comparing the performance of LDP with its counterparts (LBP and PCA). Kirsch masks in eight different directions have been used to calculate the edge response values in each block. The goal is to identify the 'k most prominent directions' and use them to form the LDP.

The results of the study have been abridged to their mean values for simpler presentation in this review. LDP texture description (80% accuracy) was found to be insensitive to illumination variations and noise. This reinforces robustness of the methodology to occlusion, and also reflects the superior performance of local directional patterns over local binary patterns (76%) and principal component analysis (54%) based methodologies. The paper also mentions that LDP produces more stable patterns even in presence of noise.

H. Local Binary Patterns combined with Local Phase Quantization

Local phase quantization (LPQ) is another descriptor for feature representation. So far, we saw methods that involve representing an image in the spatial domain, or pixel-by-pixel. In LPQ, the focus is on representing an image in the **frequency domain**, which captures the frequencies of **phases**. Phases essentially relate different positions of an image using their spatial frequencies which helps in the representation of textural information as **phase information** in the frequency domain. Local phase patterns from the phase information are quantized into a fixed set of bins to 'condense' it for computational feasibility. It is 'local' because the phase information is quantized in local neighborhoods. LPQ is a better descriptor compared to local binary patterns because it provides a 'richer' representation of features.

In the paper by B. Yuan *et al.* [27], a study was conducted that evaluates the combination of LBP and LPQ. Local binary patterns were used to extract local spatial domain features while Fourier-transform based local phase quantization was implemented to extract local frequency domain features and produce enhanced feature vectors. The study involved evaluation on the Yale face dataset [23] (images normalized to 100x80 pixels) and the AR dataset [18] (images normalized to 50x40 pixels). Concatenated LBP/LPQ histogram was used to represent the feature vectors. Nearest neighbor classifier and histogram intersection along with Chi-square distance was used for recognition using a window size of 10x8.

The results of the paper have been abridged for simpler presentation purposes. The findings from the study reflect that a broader range of features were captured by coupling LBP and LPQ (rather than implementing them in isolation), resulting in a more robust and complementary methodology for face recognition. The combination showed a higher recognition rate of 95.3%, considerably higher than using LBP (92%) and LPQ (88.4%) independently (all percentages mentioned here are averages of YALE and AR evaluation results from Table 1 [27]).

I. *Scale-Invariant Feature Transform (SIFT)*

SIFT is essentially a way of describing a local neighborhood in an image through a feature vector. The objective is to reduce an image into features as locally distinct points within it along with their description. Hence, recognition is done through comparison by trying to find the same points in other (test) images. The general idea of implementing the SIFT methodology for face recognition involves choosing a keypoint. generating a descriptor that describes local neighborhood of the keypoint and forming associations if are multiple images with the same corresponding keypoints.

It is important to understand how the SIFT descriptors are generated before delving into its application. First, the keypoints are found using difference of Gaussians approach (DoG). This involves blurring the same image using Gaussian blur at different magnitudes, and then subtracting these differently blurred images from each other which produces ‘difference images’. The difference images are ‘stacked’ which highlights the distinct points that stand out, which are nothing but the keypoints. SIFT is *scale-invariant* because the DoG is applied on different scales of the same image too (in addition to applying multiple blurs for each scale) and aggregated through the Gaussian pyramid after which keypoints are identified at different levels of the pyramid. Now that the keypoints are identified, the descriptor vector is computed by looking at the local neighborhood of the keypoints, by breaking the neighborhood down into smaller areas and computing the gradients in each of these areas. Gradients are used due to their robustness to variations. These local gradients are then represented using a histogram of frequencies of different magnitudes at different positions on the image. Hence, the SIFT descriptors are generated.

The paper of interest that demonstrates the application of using SIFT features for face recognition is a study by M. Aly *et al.* [28] that focuses on comparing the results of using SIFT-features over Eigenfaces and Fisherfaces. The evaluation methods in this study involved euclidean, city-block and cosine distances used for eigenfaces and fisherfaces, while cosine and angle distances were used to match SIFT features. Evaluation was done on the AT&T [29] and Yale [23] databases in a series of 10 experiments. Two additional experiments were also carried out with varying training (80%) and test (20%) set sizes. The number of relevant SIFT features required for reliable recognition were measured on a 50-50 train-test set. The effect of downsampling the image resolutions (25%, 50%, 75% of original resolutions) was also studied.

The findings of paper indicate that SIFT produced remarkably better average accuracy (94%) over Eigenfaces (82.5%) and Fisherfaces (90.4%) over the AT&T and Yale datasets. Even on varying training set size, SIFT consistently performed better (90.1%) than its counterparts. The key conclusion obtained from the study was that only 30% of SIFT features were needed for higher reliable recognition rates than Eigenfaces and Fisherfaces (in 91% less time). Similar trends followed in variations of resolution as well. Overall, SIFT outperforms other methodologies in most evaluation conditions and variations and also proves to be more computationally efficient.

J. FaceNet: A Unified Embedding for Face Recognition and Clustering

The research paper by F. Schroff *et al.* [30] introduces a methodology called **FaceNet**, which is based on the fundamental concept of embedding input images into Euclidean space which can capture properties of face similarity. The idea is to map the test images into an embedding space, and performing face verification and clustering. According to the paper, the training methodology involves **triplet mining**, which involves using an original image of a person (anchor image), another image of the same person (positive image) and a third image of a different person (negative image). For evaluation, L2 distance (a Euclidean distance) is used. Between a pair of points, we compare the L2 distance and recognize the faces. Proposal of this new technique eliminates role of number of classes. Learns general representation which maps any input faces to the dimensionalities while holding the properties (large-scale inputs).

In this approach, the triplet loss function is used to minimize L2 distance between anchor profile and positive profile and maximize it between anchor and negative profile. The general idea is to use an embedding function $f(x)$ (based on deep convolution model) which takes image and projects it to an embedding space such that its vector has a unit norm. Once the triplet loss function is applied, the goal is to sum all points in the embedding space such that the distance between anchor and positive profiles are minimized while keeping them as far as possible from the negative profiles. **Semi-hard triplet mining** is a common method used to mine negative profiles. This involves two key steps. For a given anchor profile A, positive profiles P lying around its space are identified. Then, a super space containing negative profiles N is drawn around the initial space, which are informative for the model even as the negative profiles are away from the inner partition but their squared distance is close enough to the anchor positive distance. Here is a visualization of this description:

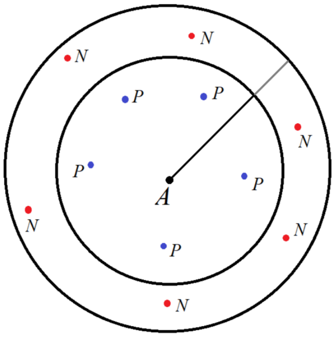

Figure 2. Illustration of Semi-hard Triplet Mining

In this paper [30], semi-hard triplet mining has been implemented in mini-batches of 1,800 input examples with anchor and positive profiles. The paper highlights that similar techniques have been previously developed but based on engineered / handmade features. However, this study uses a fully end-to-end deep learning methodology. In retrospect, the idea is to assume a case where the anchor profile has negative profiles close to it and train the learning algorithm to modify this configuration to bring anchor and positive profiles close while keep negative profile far. The model has been trained with two convolational neural networks - Zeiler & Fargus (NN1) and GoogleNet-style Inception network (NN2, NN3, NN4, NNS1, NNS2). The CNNs have been trained using Stochastic Gradient Descent (SGD) [31] with *standard backprop* [32] [33] and AdaGrad [34]. The evaluation parameters of this study are True Accept Rate (TAR), Validation Rate (VAL) and False Accept Rate (FAR). TAR compares L2 distances between profiles for a given threshold d. VAL is the True Accept ratio and FAR is False Accept ratio, computed using K-fold cross validation on $200k$ images (in $100k$ × $100k$ image pairs). Tests were also conducted on personal photos to evaluate robustness, including the use of different image resolutions and dimensions to measure change in validation rates.

The results of the paper indicate FaceNet's exceptional accuracy rates of 99.63% on the Labelled

Faces in the Wild (LFW) dataset [35] and 95.3% on the YouTube Faces DB [36]. The model was also found to be invariant to pose and illumination. Among the five models (NN1, NN2, NN3, NNS1, NNS2) tested, the Inception-based NN2 model achieved the best results. The paper also mentions a future proposal of introducing harmonic embeddings that could potentially enable higher compatibility between existing models and improve predictions.

K. Sparse Representation for Classification

Consider a high-dimensional vector, such as the millions of pixels of an image rearranged vertically into a tall million-by-one column vector. This vector can be transformed into a natural measurement space through a learned dictionary to represent the original data in terms of sparse coefficients. Let these coefficients be represented by matrix S with the same dimensions as the million-by-one matrix. S is referred to as the sparse coefficient matrix, which mostly has zero entries. Only a small number of S terms are non-zero, allowing for efficient storage and sparse representation of the data. This can be represented as: $X = D \cdot S$, where X is the original high-dimensional vector, and D is a learned dictionary that provides flexibility in representing any image as a sparse vector.

Sparse representation is used in classification (SRC) by leveraging robust statistics and patterns that exist in data through sparsity relative to a library of face images for cross-referencing. It involves some general steps. First, a person's image may be downsampled to reduce its dimensionality while maintaining essential features. It is then represented as a tall, high-dimensional vector. This process is repeated for every image in a large dataset. Either SVD or PCA can be used to reduce the dimensionality to a smaller discrete value. The objective is to obtain a sparse coefficient vector that represents a test image as a linear combination of dictionary elements. The final prediction is made by comparing the sparse coefficient vector of the test image with the coefficient vectors of known individuals to identify the closest match.

The paper of interest is the study conducted by J. Wright *et al.* [37]. It focuses on utilizing sparse representation of test images as a linear combination of training samples with $L1$-norm minimization to obtain sparse coefficient vector and then apply it for face recognition. The evaluation was done on the Extended Yale B database [12](dictionary size $\approx$ 1,200). The study compares the proposed methodology against Nearest Neighbor, Nearest Subspace, and Linear Support Vector Machines (SVM) approaches. $L1$-norm minimization was used to optimize sparse coefficient vectors. The Sparsity Concentration Index (SCI) metric determined accuracy of representing a test image using coefficients of an individual test subject (under multiple block occlusion sizes: 0%, 10%, 30%, 50%) to enhance rejection accuracy.

The results of the study indicate that SRC provides a robust recognition system with favorable accuracy (92.1%) over other feature extraction methods (like Eigenfaces, Fisherfaces) while providing comparable performance with Support Vector Machines (SVM). Harnessing sparsity and redundancy reinforces the discriminative nature for highly accurate recognition. The proposed approach is robust to most types of occlusion (lighting and expression) but not orientation, which is a key drawback.

III. Conclusion

The papers that have been reviewed reflect the ongoing development in enhancing face recognition technology. We've explored various face recognition methodologies and their evaluation criteria. While it is evident that no single methodology excels under all conditions, the right choice of methodology under necessary requirements and constraints will determine the quality of results yielded in a real-time application. For example, SIFT demonstrated exceptional accuracy and efficiency, particularly when feature dimensionality was reduced, making it a suitable option for real-time applications. Meanwhile, FaceNet proved to be invariant to occlusions such as pose and illumination, making it suitable for highly constrained environments. We also discussed the drawbacks of every approach, like the sensitivity of Eigenfaces to variations in image scaling, FaceNet potentially requiring high computational resources, and Sparse Representation-based Classification facing challenges under pose and orientation variations. Nonetheless, all methodologies reflect substantially high accuracies, produced favorable results, and their strengths always outweighed the weaknesses. We also delved into various standard datasets (FERET, AT&T, Yale, etc) used to train and test the proposed methodologies, which helped us quantify and validate the adaptability and versatility of the discussed approaches. It is important to highlight that the diversity in the discussed methodologies indicates the level of advancement that has currently been made in the field of face recognition, which continues to evolve with increasing potential for research and development as real-world requirements are becoming increasingly sophisticated.

References

[1] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision-ECCV 2004: 8th European Conference on Computer Vision, Prague, Czech Republic, May 11-14, 2004. Proceedings, Part I 8*. Springer, 2004, pp. 469–481. 2

[2] P. Phillips, H. Moon, S. Rizvi, and P. Rauss, "The feret evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090–1104, 2000. 2, 6, 7

[3] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of cognitive neuroscience*, vol. 3, no. 1, pp. 71–86, 1991. 2, 5

[4] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," *Josa a*, vol. 14, no. 8, pp. 1724–1733, 1997. 2

[5] B. Moghaddam, C. Nastar, and A. Pentland, "A bayesian similarity measure for direct image matching," in *Proceedings of 13th International Conference on Pattern Recognition*, vol. 2. IEEE, 1996, pp. 350–358. 2

[6] L. Wiskott, J.-M. Fellous, N. Krüger, and C. Von Der Malsburg, "Face recognition by elastic bunch graph matching," in *Intelligent biometric techniques in fingerprint and face recognition*. Routledge, 2022, pp. 355–396. 2

[7] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE Workshop on Applications of Computer Vision*, 1994, pp. 138–142. 2

[8] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition*. IEEE Computer Society, 1991, pp. 586–587. 3

[9] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Josa a*, vol. 4, no. 3, pp. 519–524, 1987. 3

[10] J. Ren, X. Jiang, and J. Yuan, "Relaxed local ternary pattern for face recognition," in *2013 IEEE international conference on image processing*. IEEE, 2013, pp. 3680–3684. 3

[11] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression (pie) database," in *Proceedings of fifth IEEE international conference on automatic face gesture recognition*. IEEE, 2002, pp. 53–58. 4, 6

[12] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 6, pp. 643–660, 2001. 4, 10

[13] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 684–698, 2005. 4

[14] J. Ren, X. Jiang, and J. Yuan, "A complete and fully automated face verification system on mobile devices," *Pattern Recognition*, vol. 46, no. 1, pp. 45–56, 2013. 4

[15] P. S. Prasad, R. Pathak, V. K. Gunjan, and H. Ramana Rao, "Deep learning based representation for face recognition," in *ICCCE 2019: Proceedings of the 2nd International Conference on Communications and Cyber Physical Engineering*. Springer, 2020, pp. 419–424. 5

[16] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014. 5

[17] S. K. Zhou, G. Aggarwal, R. Chellappa, and D. W. Jacobs, "Appearance characterization of linear lambertian objects, generalized photometric stereo, and illumination-invariant face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 2, pp. 230–245, 2007. 5

[18] A. Martinez and R. Benavente, "The ar face database," *[Online] CVC Technical Report# 24*, 1998. 5, 6, 8

[19] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projection," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 711–720, 1997. 5

[20] M. Anggo and L. Arapu, "Face recognition using fisherface method," in *Journal of Physics: Conference Series*, vol. 1028, no. 1. IOP Publishing, 2018, p. 012119. 5

[21] R. D. Mulyawan and C. Supriyanto, "Teknik pengenalan wajah pada database citra digital menggunakan metode eigenface," 2009. 6

[22] O. Déniz, G. Bueno, J. Salido, and F. De la Torre, "Face recognition using histograms of oriented gradients," *Pattern recognition letters*, vol. 32, no. 12, pp. 1598–1603, 2011. 6

[23] "Yale face database," http://cvc.yale.edu/projects/yalefaces/yalefaces.html, 2009, last accessed April 2009. 6, 8

[24] F. De la Torre, A. Collet, M. Quero, J. F. Cohn, and T. Kanade, "Filtered component analysis to increase robustness to local minima in appearance models," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007, pp. 1–8. 7

[25] A. Albiol, D. Monzo, A. Martin, J. Sastre, and A. Albiol,

"Face recognition using hog–ebgm," *Pattern Recognition Letters*, vol. 29, no. 10, pp. 1537–1543, 2008. 7

[26] T. Jabid, M. H. Kabir, and O. Chae, "Local directional pattern (ldp) for face recognition," in *2010 digest of technical papers international conference on consumer electronics (ICCE)*. IEEE, 2010, pp. 329–330. 7

[27] B. Yuan, H. Cao, and J. Chu, "Combining local binary pattern and local phase quantization for face recognition," in *2012 International Symposium on Biometrics and Security Technologies*. IEEE, 2012, pp. 51–53. 8

[28] M. Aly, "Face recognition using sift features," *CNS/Bi/EE report*, vol. 186, 2006. 8

[29] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE workshop on applications of computer vision*. IEEE, 1994, pp. 138–142. 8

[30] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823. 9

[31] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural networks*, vol. 16, no. 10, pp. 1429–1451, 2003. 9

[32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989. 9

[33] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986. 9

[34] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization." *Journal of machine learning research*, vol. 12, no. 7, 2011. 9

[35] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *CVPR 2011*. IEEE, 2011, pp. 529–534. 10

[36] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," in *Workshop on faces in'Real-Life'Images: detection, alignment, and recognition*, 2008. 10

[37] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2008. 10