Chapter 2

Literature Survey
This chapter gives an overview on the major human face recognition techniques, advantages and disadvantages of each method are also given. The methods considered are Principal Component Analysis, Kernel Principal Component Analysis, Linear Discriminant Analysis, Independent Component Analysis, Neural network based methods, Discrete cosine transform and Singular value decomposition.

2.1 Pattern Recognition Systems

Many pattern recognition systems can be partitioned into components such as the one shown in Figure 2.1. A sensor converts images or sounds or other physical inputs into signal data. The segmentor isolates sensed objects from background or from other objects. A feature extractor measures object properties that are useful for classification.

![Figure 2.1: Pattern Recognition System](image)

The classifier uses these features to assign the sensed object to a category. Finally, a post processor can take account of other considerations, such as the effect of context and costs of errors, to decide on appropriate action. Although this description
stresses a one-way or “bottom-up” flow of data, some systems employ feedback from higher levels back down to lower levels (gray arrows)[1].

During the past 30 years, pattern recognition has had a considerable growth. Applications of pattern recognition now include: character recognition; target detection; medical diagnosis; biomedical signal and image analysis; remote sensing; identification of human faces and of fingerprints; machine part recognition; automatic inspection; and many others.

Traditionally, Pattern recognition methods are grouped into two categories: structural methods and feature space methods. Structural methods are useful in situation where the different classes of entity can be distinguished from each other by structural information, e.g. in character recognition different letters of the alphabet are structurally different from each other. The earliest-developed structural methods were the syntactic methods, based on using formal grammars to describe the structure of an entity [2].

The traditional approach to feature-space pattern recognition is the statistical approach, where the boundaries between the regions representing pattern classes in feature space are found by statistical inference based on a design set of sample patterns of known class membership. Feature-space methods are useful in situations where the distinction between different pattern classes is readily expressible in terms of numerical measurements of this kind. The traditional goal of feature extraction is to characterize the object to be recognized by measurements whose values are very similar for objects in the same category, and very different for objects in different categories. This leads to the idea of seeking distinguishing features that are invariant to irrelevant transformations of the input. The task of the classifier component proper of a full system is to use the feature vector provided by the feature extractor to assign the object to a category. Image classification is implemented by computing the similarity score between a target discriminating feature vector and a query discriminating feature vector [3].

2.2 Face Recognition Processing

Face recognition is a visual pattern recognition problem. There, a face as a three dimensional object subject to varying illumination, pose, expression and so on is to be identified based on its two-dimensional image (three dimensional images e.g., obtained from laser may also be used). A face recognition system generally consists
of four modules as depicted in Figure 2.2: detection, alignment, feature extraction, and matching, where localization and normalization (face detection and alignment) are processing steps before face recognition (facial feature extraction and matching) is performed. *Face detection* segments the face areas from the background. In the case of video, the detected faces may need to be tracked using a *face tracking* component. *Face alignment* is aimed at achieving more accurate localization and at normalizing faces thereby whereas face detection provides coarse estimates of the location and scale of each detected face. Facial components, such as eyes, nose, and mouth and facial outline, are located; based on the location points, the input face image is normalized with respect to geometrical properties, such as size and pose, using geometrical transforms or morphing. The face is usually further normalized with respect to photometrical properties such illumination and gray scale. After a face is normalized geometrically and photometrically, *feature extraction* is performed to provide effective information that is useful for distinguishing between faces of different persons and stable with respect to the geometrical and photometrical variations. For *face matching*, the extracted feature vector of the input face is matched against those of enrolled faces in the database; it outputs the identity of the face when a match is found with sufficient confidence or indicates an unknown face otherwise.

Face recognition results depend highly on features that are extracted to represent the face pattern and classification methods used to distinguish between faces whereas face localization and normalization are the basis for extracting effective features. These problems may be analyzed from the viewpoint of face subspaces or manifolds, as follows.

![Figure 2.2: Face Recognition Processing Flow](image)

**2.3 Existing Feature Extraction Techniques**

In research of pattern recognition, we always want to achieve the correct classification rate according to the characteristics required. Feature extraction greatly
affects the design and performance of the classifier, and it is one of the core issues of PR research. As an important component of pattern recognition, feature extraction has been paid close attention by many scholars, and currently has become one of the research hot spots in the field of pattern recognition [4].

Statistical analysis theory is the frequently-used method of data feature extraction and selection. Statistical methods are based on forceful theory, have lots of algorithms, and can effectively analyze and process the data. As the representative, there’re some classical methods such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), Factor Analysis (FA), and Ordinary Least Square (OLS) and so on.

Following is list of some feature extraction algorithms used in feature extraction [5]:

<table>
<thead>
<tr>
<th>Method</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Principal Component Analysis</td>
<td>Eigenvector-based, linear map</td>
</tr>
<tr>
<td>Kernel PCA</td>
<td>Eigenvector-based, non-linear map, uses kernel methods</td>
</tr>
<tr>
<td>Linear Discriminant Analysis</td>
<td>Eigen vector-based, supervised linear map</td>
</tr>
<tr>
<td>Independent Component Analysis</td>
<td>Linear map, separate non-Gaussian distributed features</td>
</tr>
<tr>
<td>Neural Network based methods</td>
<td>Diverse neural networks using PCA, etc.</td>
</tr>
<tr>
<td>Self organizing map(SOM)</td>
<td>Nonlinear, based on a grid of neurons in the feature space</td>
</tr>
<tr>
<td>Discrete Cosine Transform</td>
<td>Linear function, Fourier-related transform, usually used 2D-DCT</td>
</tr>
<tr>
<td>Singular Value Decomposition</td>
<td>Linear Algebra</td>
</tr>
</tbody>
</table>

**2.3.1 Principal Component Analysis**

The current state of feature extraction is characterized by a family of subspace methods originated by eigenface, the core of which is principal component analysis (PCA) [6]. PCA seeks a subspace that best represents the data in the sense of mean-square error. Derived from PCA, orthogonal complement component analysis (OCCA) can deal with the problem of unbalanced positive and negative samples [7].

PCA is a kind of statistical method that turning various feature indicators to a small number of indicators that describe the data sets from the perspective of the effectiveness of the features. Suppose there are n samples, measuring each sample,
we get \( p \) indicators, then we totally have \( np \) data, however, indicators often influence each other, PCA is the study of how to find limited several compositive indicators from the \( p \) indicators (that is principal component), then achieving our purpose by analyzing principal component. Here it is requested that the principal component should reflect the information of original data as much as possible, and should be independent from each other. PCA project the high-dimensional observational data to low-dimensional master subspace under the meaning of global minimum reconstructed errors, and the subspace generated by the feature vectors that are relevant with the several largest eigenvalue of data points covariance matrix.

PCA is an effective linear dimension-reduction method that has perfect theory and feasible algorithms, but its effectiveness is built on the assumption that the data embedded in low-dimensional space that are global linear or approximate linear. PCA, which completely reserve the second-order moment information of the original data, is a kind of optimizing and simple expression. But variance cannot completely reflect the amount of information, for it can’t use the classificatory information of original data, sometimes it makes compressed data don’t conducive to pattern classification.

Existing PCA-based face recognition systems are hard to scale up because of the computational cost and memory-requirement burden. Usually, PCA is performed in batch mode. It means that all training data have to be ready for calculating the PCA projection matrix during training stage. The learning stops once the training data have been fully processed. If we want to incorporate additional training data into an existing PCA projection matrix, the matrix has to be retrained with all training data. In turn, it is hard to scale up the developed systems [8].

PCA is a classical feature extraction and data representation technique widely used in the areas of pattern recognition and computer vision. PCA aims to find a linear mapping, which preserves total variance by maximizing the trace of feature variance. The optimal mapping is the leading eigenvectors of the data’s total variance matrix associated with the leading eigenvalues. Thus, PCA preserve the total variance by maximizing the trace of feature variance, but PCA cannot preserve local information due to pursing maximal variance [9].

Principal Component Analysis (PCA) is one of the most popular techniques for dimensionality reduction. Starting from an original set of \( l \) samples (features),
which form the elements of a vector $x \in \mathbb{R}^l$, the goal is to apply a linear transformation to obtain a new set of samples:

$$ y = A^T x $$

so that the components of $y$ are uncorrelated. In a second stage, one chooses the most significant of these components. The steps are summarized here:

1. Estimate the covariance matrix $S$. Usually the mean value is assumed to be zero, $E[x] = 0$. In this case, the covariance and autocorrelation matrices coincide, $R = E[xx^T] = S$. If this is not the case, we subtract the mean. Given $N$ feature vectors, $x_i \in \mathbb{R}^l$, $i = 1, 2, \ldots, N$, the autocorrelation matrix estimate is given by

$$ R = \frac{1}{N} \sum_{i=1}^{N} x_i x_i^T $$

(1)

2. Perform the eigen decomposition of $S$ and compute the $l$ eigenvalues/eigenvectors, $\lambda_i, a_i \in \mathbb{R}^l, i = 0, 2, \ldots, l-1$.

Arrange the eigenvalues in descending order, $\lambda_0 \geq \lambda_1 \geq \cdots \geq \lambda_{l-1}$.

3. Choose the $m$ largest eigenvalues. Usually $m$ is chosen so that the gap between $\lambda_{m-1}$ and $\lambda_m$ is large. Eigenvalues $\lambda_0, \lambda_1, \ldots, \lambda_{m-1}$ are known as the $m$ principal components.

4. Use the respective (column) eigenvectors $a_i, i = 0, 1, 2, \ldots, m-1$ to form the transformation matrix $A = [ a_0 \quad a_1 \quad a_2 \quad \cdots \quad a_{m-1} ]$

5. Transform each $l$-dimensional vector $x$ in the original space to an $m$-dimensional vector $y$ via the transformation $y = A^T x$. In other words, the $i$th element $y(i)$ of $y$ is the projection of $x$ on $a_i$ ($y(i) = a_i^T x$).

The total variance of the elements of $x$, $\sum_{i=0}^{l-1} E[x^2(i)]$ (for zero mean), is equal to the sum of the eigenvalues $\sum_{i=0}^{l-1} \lambda_i$. After the transformation, the variance of the $i$th element, $E[y^2(i)]$, $i = 0, 2, \ldots, l-1$, is equal to $\lambda_i$. Thus, selection of the elements that correspond to the $m$ largest eigenvalues retains the maximum variance.

The computation of the principal components can be done as follows:

$[\text{eigenval, eigenvec,explain,}\text{Y,mean_vec}] = \text{pca\_fun}(X,m)$
where

- $X$ is an $l \times N$ matrix with columns that are the data vectors,
- $m$ is the number of the most significant principal components taken into account,
- $\text{eigenval}$ is an $m$-dimensional column vector containing the $m$ largest eigenvalues of the covariance matrix of $X$ in descending order,
- $\text{eigenvec}$ is an $l \times m$-dimensional matrix, containing in its columns the eigenvectors that correspond to the $m$ largest eigenvalues of the covariance matrix of $X$,
- $\text{explain}$ is an $l$-dimensional column vector whose $i$th element is the percentage of the total variance retained along the $i$th principal component,
- $Y$ is an $m \times N$ matrix containing the projections of the data points of $X$ to the space spanned by the $m$ vectors of $\text{eigenvec}$,
- $\text{mean}_{\text{vec}}$ is the mean vector of the column vectors of $X$ [10].

### 2.3.2 Linear Discriminant Analysis

LDA is one of the most popular linear projection techniques for feature extraction. It finds the set of the most discriminant projection vectors which can map high-dimensional samples onto a low-dimensional space. Using the set of projection vectors determined by LDA as the projection axes, all projected samples will form the maximum between-class scatter and the minimum within-class scatter simultaneously in the projective feature space. The major drawback of applying LDA is that it may encounter the so-called small sample size problem. This problem arises whenever the number of samples is smaller than the dimensionality of the samples. Under these circumstances, the sample scatter matrix may become singular, and the execution of LDA may encounter computational difficulty [11].

LDA suffer from insufficient learnability in small sample size cases, which is a problem in real applications such as face recognition. In consequence, subspaces learnt from these algorithms may have poor recognition performance on test data [12]. In the tasks of image recognition, a 2D image is usually transformed into a 1D long vector and modeled as a point in a high-dimensional vector space. This vector-space model brings up much convenience and many advantages. However, it also leads to some problems such as the Curse of Dimensionality dilemma and Small Sample Size problem, and thus produces us a series of challenges, for example, how to deal with the problem of numerical instability in image recognition, how to improve the
Development of Feature Extraction Techniques for Face Recognition

accuracy and meantime to lower down the computational complexity and storage requirement in image retrieval, and how to enhance the image quality and meanwhile to reduce the transmission time in image transmission, etc [13].

LDA is used to find the optimal set of projection vectors that maximize the determinant of the between-class scatter matrix and at the same time minimize the determinant of the within-class scatter. But, the dimension of vectors is high and the number of observations is small, usually tens or hundreds of samples. An intrinsic limitation of traditional LDA is that it fails to work when the within-class scatter matrix becomes singular. This is known as the small sample size (SSS), the under sampled or singularity problem. Furthermore, class discrimination in LDA is based upon interclass and intraclass scatters, which is optimal only in cases where the data of each class is approximately Gaussian distributed, a property that cannot always be satisfied in real-world applications.

LDA is a supervised learning algorithm. Let $l$ denote the total class number and $l_i$ denote the number of training samples in the $i$th class. Let $x_{ij}$ denote the $j$th sample in $i$th class, $\bar{x}$ be the mean of all the training samples $\bar{x}_i$ be the mean of the $i$th class. The between-class and within-class scatter matrices can be evaluated by:

$$S_b = \sum_{i=1}^{l} l_i (\bar{x}_i - \bar{x})(\bar{x}_i - \bar{x})^T$$ (2)

$$S_w = \sum_{i=1}^{l} \sum_{j=1}^{l_i} (x_{ij} - \bar{x}_i)(x_{ij} - \bar{x}_i)^T$$ (3)

LDA aims to find an optimal projection $U_{opt}$ such that the ratios of the between-class scatter to within-class scatter is maximized, i.e.

$$U_{opt} = \arg \max (U) \left| \frac{U^T S_b U}{U^T S_w U} \right| = [u_1, u_2, \ldots, u_d]$$ (4)

where $\{u_i \mid i = 1, 2, \ldots, d\}$ is the set of generalized eigenvectors of $S_b$ and $S_w$ corresponding to the $d$ largest generalized eigen values $\{\lambda_i \mid i = 1, 2, \ldots, d\}$, i.e.

$$S_b u = \lambda_i S_w u, \quad i = 1, 2, \ldots, d.$$ (5)

Note that there are at most $c - 1$ non-zero generalized eigen values.

Belhumeur et al. switched from eigenface to Fisherface [14]. Fisherface consists of two stages: PCA followed by linear discriminant analysis (LDA). Utilizing discrimination criterion, LDA defines a projection that makes the within-class scatter
small and the between-class scatter large. By inexplicitly nonlinear mapping input space to a high dimensional space using kernel function, kernel PCA (KPCA) [15] and kernel LDA (KLDA) can extract nonlinear features [16]. The above methods can extract low-dimensional features from the intensity of face images. More robust features can be obtained from the Gabor coefficients of face images. Elastic bunch graph matching (EBGM) describes faces using Gabor filter responses in certain facial landmarks and a graph describing the spatial relations of these landmarks [17]. Instead of extracting Gabor features from the facial landmarks, appearance-based methods compute Gabor features from each pixel. However, the dimensionality of the augmented Gabor feature vector space is too high. To reduce the dimensionality, Liu et al. applied PCA (or LDA) on the augmented Gabor feature vector, which we call Gabor PCA, and Gabor LDA, respectively [18].

Within the past decade, major advances have occurred in face recognition. Numerous algorithms have been proposed for face recognition. These methods have a common characteristic in finding low dimension feature subspace lying in high-dimensional face space. Two of the most classical subspace methods are principal component analysis (PCA) and linear discriminant analysis (LDA). PCA is a method that reduces data dimensionality by performing a covariance analysis between factors, and then projects the data along the directions where the data vary the most. These directions are determined by the eigenvectors of the covariance matrix corresponding to the largest eigenvalues. The set of eigenvectors is named Eigenfaces, which is then used to describe face images. LDA is a supervised learning method. It maximizes the ratio of between-class variance to the within-class variance in any particular data set thereby guaranteeing maximal separability. The prime difference between LDA and PCA is that PCA does more of feature classification and LDA does data classification. In PCA, the shape and location of the original data sets are changed when transformed to a different space whereas LDA does not change the location but only tries to provide more class separability. The approach using LDA to describe face images is called Fisherfaces [19].

2.3.3 The Kernel Principal Component Analysis

A subspace of low dimension is first constructed as, for example, the span of the $m$ dominant directions in the original $\mathbb{R}^l$, $l > m$ space.
The choice of dominant directions depends on the method used. In a second stage, all vectors of interest in $\mathbb{R}^d$ are (linearly) projected in the low-dimensional subspace. Such techniques are appropriate whenever our data in $\mathbb{R}^d$ lie (approximately) on a linear manifold (e.g., hyperplane).

Figure 2.3: (a) The 2-dimensional data points lying (approximately) on a line (linear manifold). (b) The 2-dimensional data points lying (approximately) on a semicircle (nonlinear manifold).

However, in many cases the data are distributed around a lower-dimensional manifold, which is not linear (e.g., around a circle or a sphere in a 3-dimensional space).

Figures 2.3(a,b) show two cases where data in the 2-dimensional space lie (approximately) on a linear and a nonlinear manifold, respectively. Both manifolds are 1-dimensional since a straight line and the circumference of a circle can be parameterized in terms of a single parameter.

The kernel PCA is one technique for dimensionality reduction when the data lie (approximately) on a nonlinear manifold. According to the method, data are first mapped into a high-dimensional space via a nonlinear mapping:

$$ x \in \mathbb{R}^d \rightarrow \varphi(x) \in H $$

PCA is then performed in the new space $H$, chosen to be an RKHS. The inner products can be expressed in terms of the kernel trick.

Although a (linear) PCA is performed in the RKHS space $H$, because of the nonlinearity of the mapping function $\varphi(\cdot)$, the method is equivalent to a nonlinear function in the original space. Moreover, since every operation can be expressed in inner products, the explicit knowledge of $\varphi(\cdot)$ is not required. All that is necessary is to adopt the kernel function that defines the inner products.
We can use the kernel PCA as follows:

\[ [s, V, Y] = kernel\_PCA(X, m, choice, para) \]

where

- \( X \) is an \( l \times N \) matrix whose columns contain the data vectors,
- \( m \) is the number of (significant) principal components that will be considered,
- \( choice \) is the type of kernel function to be used (‘pol’ for polynomial, ‘exp’ for exponential),
- \( para \) is a 2-dimensional vector containing the parameters for the kernel function; for polynomials it is \( (x^T \ y + para(1))^{para(2)} \) and for exponentials it is \( \exp(-(x - y)^T (x - y)/(2para(1)^2)) \),
- \( s \) is an \( N \)-dimensional vector that contains the computed eigenvalues after applying the kernel PCA,
- \( V \) is an \( N \times N \) matrix whose columns are the eigenvectors corresponding to the principal components of the Gram matrix, \( K \), which is involved in the kernel PCA,
- \( Y \) is an \( m \times N \) dimensional matrix that contains the projections of the data vectors of \( X \) on the subspace spanned by the \( m \) principal components.

### 2.3.4 Independent Component Analysis

Independent Component Analysis (ICA) is a kind of new statistical method developed in recent years, the goal of the method is that the observation data will be processed with some kind of linear decomposition, so as to make the data be decomposed to statistical independent components. The basic idea of ICA is to use a hidden statistical model \( x = As \). This statistical model is called Independent Component Analysis, that is ICA model; it represents how the observation data are produced by mixing of independent components. Independent Components are hidden variables, this means they can’t be observed directly, and mixing matrix is supposed unbeknown. What can be observed is only random vector \( x \), \( A \) and \( s \) must be estimated, and it must be estimated under the assumed conditions as few as possible. ICA supposes that components are statistical independent, and the independent components are must supposed to be not Gauss distribution, unbeknown mixed matrix are supposed to be square matrix, if the inverse matrix of \( A \) can be ciphered out, assumed as \( W \), then the independent components can be got by

\[ S = Wx \]  \hspace{1cm} (6)
It can be known that there are two uncertainties of ICA model: (i) not sure of the variance of independent components; (ii) cannot ensure the order of the independent components. Literature used the principle of ICA to extract pattern feature, literature used ICA to do research on face recognition, these research show reliable and feasible methods of feature compression, then process the data features, and this will enable the recognition accuracy higher.

The main significance of feature extraction lies in the “Low Loss Dimensionality Reduction”, which enable the problems tend to be simplified and can be easily computed, or increase computation speed then the learning and training of the system becomes easy.

The traditional ICA method has limitations on non-linear problems during facial feature extraction process. Kernel function defines the vector of inner product in the feature space, and makes it possible for non-linear algorithm through a linear algorithm by the introduction of pre-selected kernel function. The data space was mapped to high dimension feature space by kernel function [20].

2.3.5 Neural Network Based Methods

Artificial neural networks are being applied to a wide variety of automation problems including adaptive control, optimization, medical diagnosis, decision making, as well as information and signal processing, including speech processing. ANNs have proven to be very suitable for processing sensor data, in particular, feature extraction and automated recognition of signals and multi-dimensional objects.

Pattern recognition has, however, emerged as a major application because the network structure is suited to tasks that biological systems perform well, and pattern recognition is a good example where biological systems out-perform traditional rule-based artificial intelligence approaches.

Today there are two classes of ANN paradigm, supervised and unsupervised. The multilayer back-propagation network (MLBPN) is the most popular example of a supervised network. It results from work carried out in the mid-eighties largely by David Rumelhart and David Parker.

It is a very powerful technique for constructing nonlinear transfer functions between several continuous valued inputs and one or more continuously valued
outputs. The network basically uses multilayer perceptron architecture and gets its name from the manner in which it processes errors during training [21].

2.3.6 Self-Organizing Maps

Self-organizing maps are instances of the so-called competitive neural network models. They are composed of a layer of input units fully connected to a layer of output units, the latter units being organized in a particular topology, such as a ring structure. These models “self-organize” (without any external supervision) through an iterative adjustment of their connection weights to find some regularity or structure in the input data. They are typically used to categorize or cluster data [21].

2.3.7 Fourier Transform and Discrete Cosine Transform

The spatial-frequency techniques such as Fourier transform [22], Discrete Cosine Transform (DCT) [23], etc. are useful in extracting the facial features at some preferred frequency. In these methods, firstly the images are transformed to the frequency domain, and thereafter, the coefficients of low frequency band are taken as the invariant image features. Furthermore, the moment invariants are the most widely used image descriptors in many pattern recognition applications such as character recognition, palm print verification, etc [24].

Although the Fourier transform of the entire time series does contain information about the spectral components in time series, it cannot detect the time distribution of different frequency, so for a large class of practical applications, the Fourier transform is unsuitable [25].

2.3.8 Singular Value Decomposition

SVD is a powerful linear algebra technique for solving linear-equations in the least-square sense, and works even for singular matrices or matrices numerically close to singular. SVD proceeds by noting that m x n matrix A, m>=n can be decomposed into a product of three matrices,

\[ A = UDV^T \]  

(7)

In which U has orthonormal columns, D is non-negative diagonal, and V^T has orthonormal rows. SVD can be used to find a solution of a set of linear equations corresponding to a singular matrix that has no exact solution it locates the closest possible solution in a least-square sense [26]. The advantage of SVD over eigen
decomposition that for a prescribed accuracy of computation SVD requires half the numerical precision of eigen decomposition [27].

SVD has the disadvantage that it is not fast from the computational point of view, and the problem of which its application is strongly conditional due to excessive work of associate calculations [28]. Another problem with the classic SVD method for object recognition suffers from the disadvantage of unsupervised learning and could not obtain high recognition accuracy [29].

2.3.9 The other methods

Pattern feature extraction and selection is the active research field all along, many scholars have made great efforts and also made great contribution, in addition to the improved methods that based on the classical methods, and many scholars have put forth some new theoretical methods such as Projection Pursuit (PP), Manifold Learning (ML) and so on.

PP is a kind of statistical method used to analyze and process high-dimensional observation data, especially to process the high-dimensional data from not normal distribution, its basic idea is to project high-dimensional data to \((1 - 3)\) subspace, seek the projection that can reflect the structure or pattern of high-dimensional data, so as to achieve the purpose of studying and analyzing high-dimensional data. This method is not bound by the assumptions of normal distribution, and in reality there are many data that are not consonant with the normal distribution; people do not have enough transcendental knowledge of the data distribution; this method overcomes the problems brought about by “dimension disaster”, at the same time increases the visibility of the data; it can exclude variables interference that have nothing to do with or have little relationship with data structure and features.

A manifold can be simply thought as a topological space, it is locally Euclidean. ML generally refers to a class of unsupervised statistical learning problems, and its main goal is to find low-dimensional smooth manifold embedded in a high-dimensional observation data space, the research content of ML mainly includes the dimensionality reduction of limited data sets that reserve or highlight special features of the original data; density estimation problems of high-dimensional limited sample points that submitted to a distribution; establishing hidden variable model of high-dimensional observation data that influenced by a small number of
potential factors. ML can be divided into methods based on local and global, sometimes it can be divided into spectral methods and non-spectral methods.

Many scholars have raised many methods in data feature extraction, and tested and verified them in applications. In addition to that has been discussed above, there are still many other methods, such as Hyperplane Data Compression, Noise Suppression and Extraction Combination Features; traditional Fisher Linear Discriminant method; Wavelet Multi resolution Decomposition method for feature extraction, etc.

2.4 Face Classification

Once the features are extracted and selected, the next step is to classify the image. Sometimes two or more classifiers are combined to achieve better results. On the other hand, most model-based algorithms match the samples with the model or template. Then, a learning method can be used to improve the algorithm. One way or another, classifiers have a big impact in face recognition. According to Jain, Duin and Mao, there are three concepts that are key in building a classifier - similarity, probability and decision boundaries. We will present the classifiers from that point of view.

2.4.1 Similarity

This approach is intuitive and simple. Patterns that are similar should belong to the same class. This approach has been used in the face recognition algorithms implemented later. The idea is to establish a metric that defines similarity and a representation of the same-class samples. For example, the metric can be the euclidean distance. The representation of a class can be the mean vector of all the patterns belonging to this class. The 1-NN decision rule can be used with this parameters. Its classification performance is usually good. This approach is similar to a k-means clustering algorithm in unsupervised learning. There are other techniques that can be used [30].

<table>
<thead>
<tr>
<th>Method</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Template matching</td>
<td>Assign sample to most similar template</td>
</tr>
<tr>
<td>Nearest Mean</td>
<td>Assign pattern to nearest class mean</td>
</tr>
<tr>
<td>Subspace Method</td>
<td>Assign pattern to nearest class subspace</td>
</tr>
<tr>
<td>1-NN</td>
<td>Assign pattern to nearest pattern’s class</td>
</tr>
<tr>
<td>k-NN</td>
<td>Like 1-NN, but assign to the majority of k nearest patterns</td>
</tr>
</tbody>
</table>

Table 2.2: Similarity based Classifiers
2.4.2 Probability

Some classifiers are build based on a probabilistic approach. Bayes decision rule is often used. The rule can be modified to take into account different factors that could lead to miss-classification. Bayesian decision rules can give an optimal classifier, and the Bayes error can be the best criterion to evaluate features. Therefore, a posteriori probability functions can be optimal.

<table>
<thead>
<tr>
<th>Method</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bayesian</td>
<td>Assign pattern to the class with the highest estimated posterior probability</td>
</tr>
<tr>
<td>Logistic Classifier</td>
<td>Predicates probability using logistic curve method</td>
</tr>
<tr>
<td>Parzen Classifier</td>
<td>Bayesian classifier with Parzen density estimates</td>
</tr>
</tbody>
</table>

2.4.3 Decision boundaries

This approach can become equivalent to a Bayesian classifier. It depends on the chosen metric. The main idea behind this approach is to minimize a criterion (a measurement of error) between the candidate pattern and the testing patterns. One example is the Fisher’s Linear Discriminant. It’s closely related to PCA.

FLD attempts to model the difference between the classes of data, and can be used to minimize the mean square error or the mean absolute error. Other algorithms use neural networks. Multilayer perceptron is one of them. They allow nonlinear decision boundaries. However, neural networks can be trained in many different ways, so they can lead to diverse classifiers. They can also provide a confidence in classification, which can give an approximation of the posterior probabilities. See Table 2.4 for some decision boundary-based methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fisher Linear Discriminant (FLD)</td>
<td>Linear Classifier. Can use MSE optimization</td>
</tr>
<tr>
<td>Binary Decision Tree</td>
<td>Nodes are features. Can use FLD. Could need pruning</td>
</tr>
<tr>
<td>Perceptron</td>
<td>Iterative optimization of a classifier (e.g. FLD)</td>
</tr>
<tr>
<td>Multi-layer Perceptrons</td>
<td>Two or more layers. Uses sigmoid transfer function</td>
</tr>
<tr>
<td>Radial Basis Network</td>
<td>Optimization of a Multilayer perceptron</td>
</tr>
<tr>
<td>Support Vector Machines</td>
<td>Maximizes margin between two classes</td>
</tr>
</tbody>
</table>
2.5 Application of Face Recognition

As one of the most successful applications of image analysis and understanding, face recognition has recently received significant attention, especially during the past few years. This is evidenced by the emergence of face recognition conferences such as the International Conference on Audio and Video-Based Authentication (AVBPA) since 1997 and the International Conference on Automatic Face and Gesture Recognition (AFGR) since 1995, systematic empirical evaluations of face recognition techniques (FRT), including the FERET, FRVT 2002, and XM2VTS protocols, and many commercially available systems.

There are at least two reasons for this trend; the first is the wide range of commercial and law enforcement applications and the second is the availability of feasible technologies. The strong need for user-friendly systems that can secure our assets and protect our privacy without losing our identity in a sea of numbers is obvious. At present, one needs a PIN to get cash from an ATM, a password for a computer, a dozen others to access the internet, and so on.

Typical Applications of Face Recognition Areas Specific applications Video game, virtual reality, training programs, Entertainment Human-robot-interaction, human-computer-interaction, Drivers’ licenses, entitlement programs, Smart cards Immigration, national ID, passports, voter registration, Welfare fraud TV Parental control, personal device logon, desktop logon Information security, Application security, database security, file encryption, Intranet security, internet access, medical records Secure trading terminals Law enforcement Advanced video surveillance, CCTV control and surveillance Portal control, post event analysis, Shoplifting, suspect tracking and investigation example, fingerprint analysis and retinal or iris scans, these methods rely on the cooperation of the participants, whereas a personal identification system based on analysis of frontal or profile images of the face is often effective without the participant’s cooperation or knowledge. Table 2.5 lists some of the applications of face recognition.

Commercial and law enforcement applications of FRT range from static, controlled-format photographs to uncontrolled video images, posing a wide range of technical challenges and requiring an equally wide range of techniques from image processing, analysis, understanding, and pattern recognition. One can broadly classify FRT systems into two groups depending on whether they make use of static images or
of video. Within these groups, significant differences exist, depending on the specific application. The differences are in terms of image quality, amount of background clutter (posing challenges to segmentation algorithms), variability of the images of a particular individual that must be recognized, availability of a well-defined recognition or matching criterion, and the nature, type, and amount of input from a user.

Table 2.5: Typical Applications of Face Recognition

<table>
<thead>
<tr>
<th>Areas</th>
<th>Specific Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biometrics</td>
<td>Driver’s Licenses, Entitlement programs</td>
</tr>
<tr>
<td></td>
<td>Immigration, National ID, Passports, Voter Registration</td>
</tr>
<tr>
<td></td>
<td>Welfare Fraud</td>
</tr>
<tr>
<td>Information Security</td>
<td>Desktop Logon (Windows NT, Windows 95)</td>
</tr>
<tr>
<td></td>
<td>Application Security, Database Security, File Encryption</td>
</tr>
<tr>
<td></td>
<td>Intranet Security, Internet Access, Medical Records</td>
</tr>
<tr>
<td></td>
<td>Secure Trading Terminals</td>
</tr>
<tr>
<td>Law Enforcement and Surveillance</td>
<td>Advanced Video Surveillance, CCTV Control</td>
</tr>
<tr>
<td></td>
<td>Portal Control, Post-Event Analysis</td>
</tr>
<tr>
<td></td>
<td>Shoplifting and Suspect Tracking and Investigation</td>
</tr>
<tr>
<td>Smart Cards</td>
<td>Stored Value Security, User Authentication</td>
</tr>
<tr>
<td>Access Control</td>
<td>Facility Access, Vehicular Access</td>
</tr>
</tbody>
</table>
REFERENCES


[5] Website:


[25] Website:


[28] Website:
http://www.lokminglui.com/CaoSVDintro.pdf

[29] Website:
http://www.cbsr.ia.ac.cn/publications/yhwang/Face%20Verification%20Based%20on%20Singular%20Value%20Decomposition%20and%20Radial%20Basis%20Function%20Neural%20Network.pdf

[30] Website: