Indian Institute of Information Technology, Allahabad

Project Report
on
Face Recognition

Submitted to :

Dr. Triloki Pant
(Information Technology)

Submitted By:

Gurtej Singh
(MIT2018068)

Abstract

Face (facial) recognition is the identification of humans by the unique characteristics of their faces. Face recognition technology is the least intrusive and fastest bio-metric technology. It works with the most obvious individual identifier the human face. With increasing security needs and with advancement in technology, extracting information has become much simpler. This project aims on building an application based on face recognition using different algorithms and comparing the results. The basic purpose being to identify the face and retrieving information stored in database. It involves two main steps. First to identify
the distinguishing factors in image and storing them and second step to compare it with the existing images and returning the data related to that image. The algorithm used for face detection are PCA Algorithm.

Keywords : Face Recognition, PCA Algorithm, Eigenfaces

Introduction

1. **Biometrics**

   Biometrics is used in the process of authentication of a person by verifying or identifying that a user requesting a network resource is who he, she, or it claims to be, and vice versa.It uses the property that a human trait associated with a person itself like structure of data with the incoming data we can verify the identity of a particular person. There are many types of biometric system like detection and recognition, iris recognition etc., these traits are used for human identification in surveillance system, criminal identification , face details etc. By comparing the existing fingerprint recognition.

2. **Face Recognition**

   Security and authentication of a person is a crucial part of any industry. There are many techniques used for these purpose one of them is face recognition. Face recognition is an effective means of authenticating a person the advantage of this approach is that, it enables us to detect changes in the face pattern of an individual to an appreciable extent the recognition system can tolerate local variations in the face expressions of an individual. Hence face recognition can be used as a key factor in crime detection mainly to identify criminals there are several approaches to face recognition of which principal component(PCA) and neural networks have been incorporated in our project face recognition as many applicable areas. Moreover it can be categories into face recognition, face classification, one, or sex determination. The system consist of a database of a set of facial patterns for each individual. The characteristic features called ‘eigen faces’ are extracted from the storage images using which the system is trained for subsequent recognition of new images.

Advancements in computing capability over the past few decades have en-abled comparable recognition capabilities from such engineered systems quite successfully.Early face recognition algorithms used simple geometric models, but recently the recognition process has now matured into a science of sophisticated mathematical representations and matching processes. Major advancements and initiatives have propelled face recognition technology into the spotlight.

Face recognition technology can be used in wide range of applications. Computers that detect and recognize faces could be applied to a wide variety of practical applications including criminal identification etc. Face detection and recognition is used in many places nowadays,verifying websites hosting images and social networking sites. Face recognition and detection can be achieved using technologies related to computer science. Features extracted from a face are processed and compared with similarly processed faces present in the database. If a face is recognized it is known or the system may show a similar face existing in database else it is unknown. In surveillance system if a unknown face appears more than one time then it is stored in database for further recognition. These steps are very useful in criminal identification. In general, face recognition techniques can be divided into two groups based on the face representation they use appearance-based, which uses holistic texture features and is applied to either whole-face or specific face image and feature-based, which uses geometric facial features (mouth, eyebrows, cheeks etc), and geometric relationships between them. (A few example applications are shown in Fig 1)

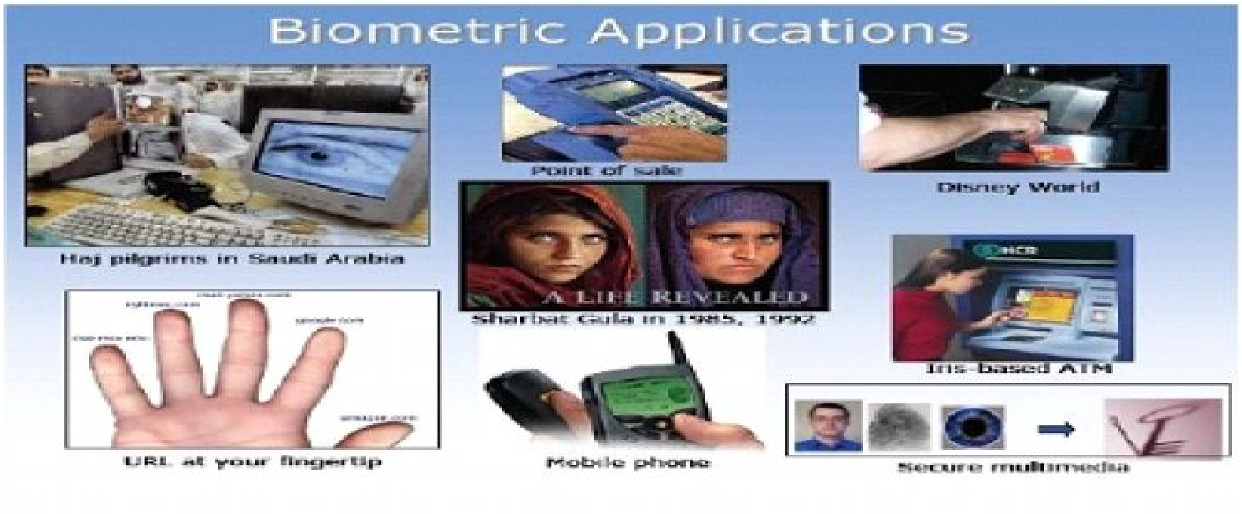


Figure 1 : Biometric Applications

In Fig 1 An important aspect is that such technology should be able to deal with various changes in face images, like rotation, changes in expression. Surprisingly, the mathematical variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to changes in face identity. This presents a great challenge to face recognition. At the core,two issues are central to successful face recognition algorithms: First, the choice of features used to represent a face. Since images are subject to changes in viewpoint, illumination, and expression, an eﬀective representation should be able to deal with these possible changes.

Secondly, the classification of a new face image using the chosen representation. Face Recognition can be of two types:

Feature Based (Geometric)

Template Based (Photometric)

In geometric or feature-based methods, facial features such as eyes, nose, mouth, and chin are detected. Properties and relations such as areas, distances, and angles between the features are used as descriptors of faces. Although this class of methods is economical and efficientin achieving data reduction and is insensitive to variations in illumination and viewpoint, it relies heavily on the extraction and measurement of facial features. Unfortunately, feature extraction and measurement techniques and algorithms developed to date have not been reliable enough to cater to this need. In contrast, template matching and neural methods generally operate directly on an image-based representation of faces, i.e., pixel intensity array. Because the detection and measurement of geometric facial features are not required,this type of method has been more practical and easier to implement when compared to geometric feature-based methods.

Background

Human beings have recognition capabilities that are unparalleled in the modern computing era. These are mainly due to the high degree of interconnectivity, adaptive nature,learning skills and generalization capabilities of the nervous system. The human brain has numerous highly interconnected biological neurons which, on some specific tasks, can outperform supercomputers. A child can accurately identify a face, but for a computer it is a cumbersome task. Therefore, the main idea is to engineer a system which can emulate what a child can do.

- **EXISTING SYSTEM**
    - Face recognition biometrics is the science of programming a computer to recognize a human face. When a person is enrolled in a face recognition system, a video camera takes a series of snapshots of the face and then represents it by a unique holistic code.
    - When someone has their face verified by the computer, it captures their current appearance and compares it with the facial codes already stored in the system.
    - The faces match, the person receives authorization; otherwise, the person will not be identified. The existing face recognition system identifies only static face images that almost exactly match with one of the images stored in the database.
    - When the current image captured almost exactly matches with one of the images stored then the person is identified and granted access.
    - When the current image of a person is considerably different, say, in terms of facial expression from the images of that person which are already stored in the database the system

does not recognize the person and hence access will be denied

- **LIMITATIONS OF THE EXISTING SYSTEM**
  - The existing or traditional face recognition system has some limitations which can be overcome by adopting new methods of face recognition :
    - The existing system cannot tolerate variations in the new face image. It requires the new image to be almost exactly matching with one of the images in the database which will otherwise result in denial of access for the individual.
    - The performance level of the existing system is not appreciable.
    - The database stored is huge, in terms of bytes in the memory.
    - Hence, processing becomes slow.
- **Proposed System and Its Advantages**

  The proposed face recognition system overcomes certain limitations of the existing face recognition system. It is based on extracting the dominating features of a set of human faces stored in the database and performing mathematical operations on the values corresponding to them. Hence when a new image is fed into the system for recognition the main features are extracted and computed to find the distance between the input image and the stored images. Thus, some variations in the new face image to be recognized can be tolerated. When the new image of a person differs from the images of that person stored in the database, the system will be able to recognize the new face and identify who the person is. The proposed system is better mainly due to the use of facial features rather than the entire face. Its advantages are in terms of
  - Computational cost because smaller images (main features) require less processing to train the PCA.

Methodology

The previous section tells us that loads of data in the memory has been a limitation in existing technologies. different techniques and methods of face detection and recognition. I will try to propose a method that overcomes this limitation. Systems with robustness and certain level of accuracy are still far away.

Keeping in view, the following architecture is proposed for the detection and recognition system. As discussed earlier that the robust system catering the needs of real world situation is a challenging task. The images will be stored into database. Again the image of the same person will be stored into the database. The first step is to select desired images from the database then for comparisons them the next step is to detect faces from each image. The next step is to recognize that images as of the same candidate or not.

Principal Component Analysis

In statistics, principal components analysis (PCA) is a technique that can be used to simplify a dataset. It is a linear transformation that chooses a new coordinate system for the data set such that the greatest variance by any projection of the data set comes to lie on the first axis (called the first principal component), the second greatest variance on the second axis, and so on. PCA can be used for reducing dimensionality in a dataset while retaining those characteristics of the dataset that contribute most to its variance, by keeping lower-order principal components and ignoring higher-order ones. The idea is that such low-order components often contain the "most important" aspects of the data.

The task of facial recognition is discriminating input signals (image data) into several classes (persons). The input signals are highly noisy (e.g. the noise is caused by differing lighting conditions, pose etc.), yet the input images are not completely random and in spite of their differences there are patterns which occur in any input signal. Such patterns, which can be observed in all signals could be - in the domain of facial recognition - the presence of some objects (eyes, nose, mouth) in any face as well as relative distances between these objects. These characteristic features are called eigenfaces in the facial recognition domain (or principal components generally). They can be extracted out of original image data by means of the mathematical tool called Principal Component Analysis (PCA).

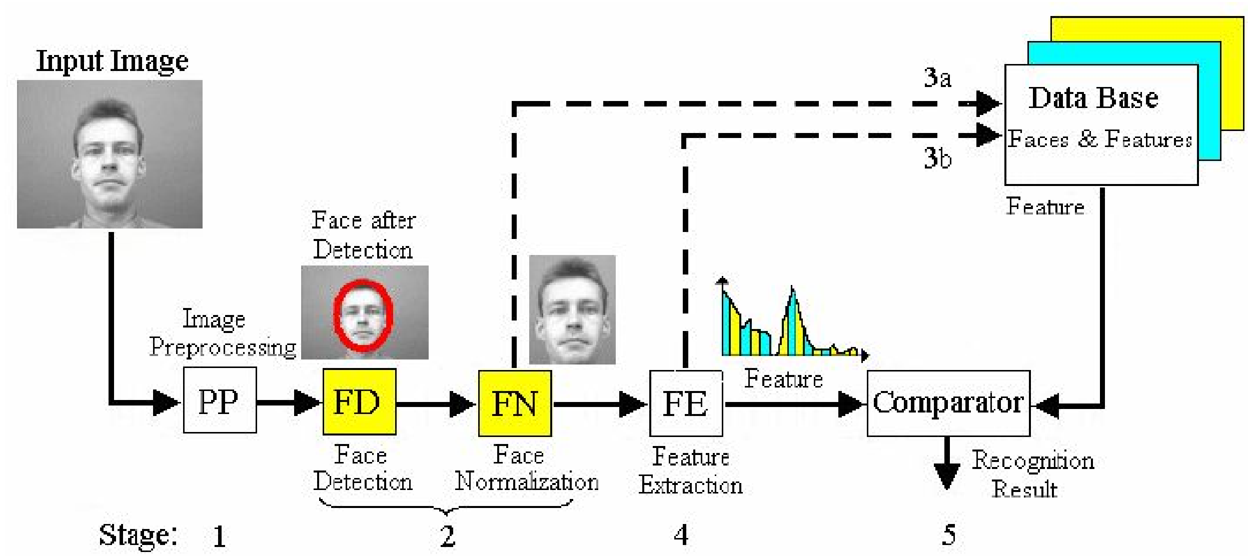


Figure 2: Structure of Face Recognition System

By means of PCA one can transform each original image of the training set into a corresponding eigenface. original image. If one uses all the eigenfaces extracted from original images, one can reconstruct the original images from the eigenfaces exactly. But one can also use only a part of the eigenfaces. Then the reconstructed image is an approximation of the original image. However, losses due to omitting some of the eigenfaces can be minimized. This happens by choosing only the most important features (eigenfaces).

Omission of eigenfaces is necessary due to scarcity of computational resources. Thus the purpose of PCA is to reduce the large dimensionality of the face space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a

strong correlation between observed variables. To generate a set of eigenfaces, a large set of digitized images of human faces, taken under the same lighting conditions, are normalized to line up the eyes and mouths. They are then all resample at the same pixel resolution (say m×n), and then treated as mn-dimensional vectors whose components are the values of their pixels. The eigenvectors of the covariance matrix of the statistical distribution of face image vectors are then extracted. Since the eigenvectors belong to the same vector space as face images, they can be viewed as if they were m×n pixel face images: hence the name eigenfaces. Viewed in this way, the principal eigenface looks like a bland androgynous average human face. Some subsequent eigenfaces can be seen to correspond to generalized features such as left-right and top-bottom asymmetry, or the presence or lack of a beard. Other eigenfaces are hard to categorize, and look rather strange. When properly weighted, eigenfaces can be summed together to create an approximate gray scale rendering of a human face.
Remarkably few eigenvector terms are needed to give a fair likeness of most people's faces, so eigenfaces provide a means of applying data compression to faces for identification purposes. It is possible not only to extract the face from eigenfaces given a set of weights, but also to go the opposite way. This opposite way would be to extract the weights from eigenfaces and the face to be recognized. These weights tell nothing less, as the amount by which the face in question differs from "typical" faces represented by the eigenfaces. Therefore, using this weights one can determine two important things:

- Determine if the image in question is a face at all. In the case the weights of the image differ too much from the weights of face images (i.e. images, from which we know for sure that they are faces) the image probably is not a face.
- Similar faces (images) possess similar features (eigenfaces) to similar degrees (weights). If one extracts weights from all the images available, the images could be grouped to clusters. That is, all images having similar weights are likely to be similar faces.

EIGENVALUES AND EIGENVECTORS

Large matrices can be costly, in terms of computational time, to use. Large matrices may have to be iterated hundreds or thousands of times for a calculation. Additionally, the behavior of matrices would be hard to explore without important mathematical tools. One mathematical tool, which has applications not only for Linear Algebra but for differential equations, calculus, and many other areas, is the concept of eigenvalues and eigenvectors.

The words eigenvalue and eigenvector are derived from the German word "eigen" which means "proper" or "characteristic." An eigenvalue of a square matrix is a scalar that is usually represented by the Greek letter λ and an eigenvector is a non-zero vector denoted by the small letter x.

For a given square matrix, A, all eigenvalues and eigenvectors satisfy the equation $Ax = \lambda x$ In other words, an eigenvector of a matrix is a vector such that, if multiplied with the matrix, the result is always an integer multiple of that vector. This integer value is the corresponding eigenvalue of the eigenvector.

Eigenvectors possess following properties

- They can be determined only for square matrices
- There are n eigenvectors in a $n \times n$ matrix.
- All eigenvectors are perpendicular, i.e. at right angle with each other. Since each eigenvector is associated with an eigenvalue, we often refer to an x and λ that correspond to one another as an eigenpair.
- An eigenspace is a space consisting of all eigenvectors which have the same eigenvalue. These eigenvectors are derived from the covariance matrix of the probability distribution of the high-dimensional vector space of possible faces of human beings and hence eigenfaces are a set of eigenvectors.

PRINCIPAL COMPONENTS

Principal Components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (i.e., principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. So, the idea is 10-dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on, until having something like shown in the scree plot below.

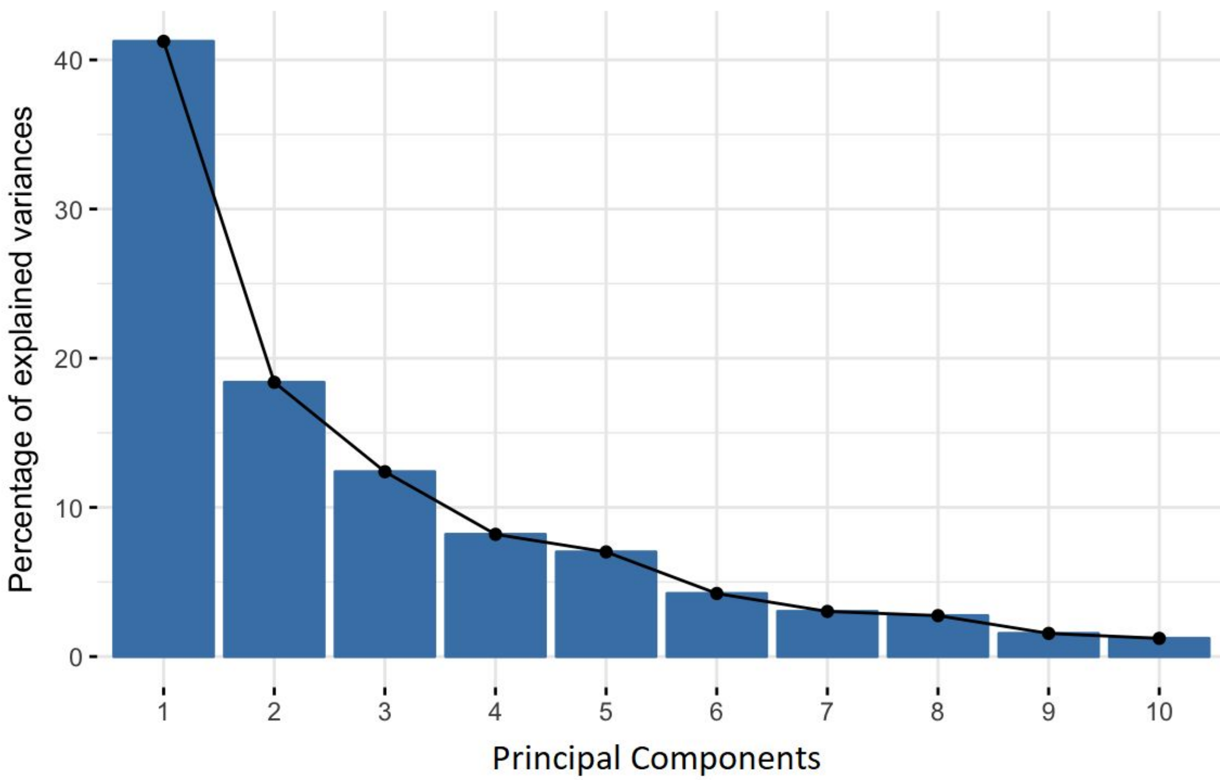


Figure 3 : Percentage of variance (information) for by each PC

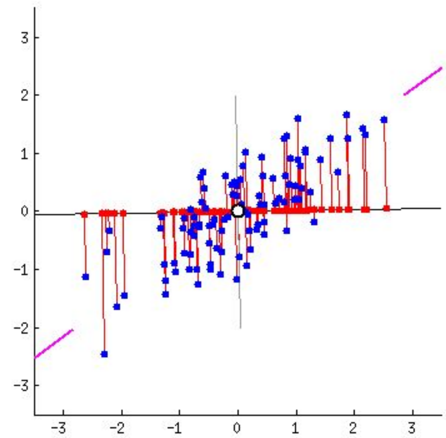
Organizing information in principal components this way, will allow you to reduce dimensionality without losing much information, and this by discarding the components with low information and considering the remaining components as your new variables.

An important thing to realize here is that, the principal components are less interpretable and don't have any real meaning since they are constructed as linear combinations of the initial variables.

Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data. The relationship between variance and information here, is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more the information it has. To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.

How PCA constructs the Principal Components?

As there are as many principal components as there are variables in the data, principal components are constructed in such a manner that the first principal component accounts for the largest possible variance in the data set. For example, let's assume that the scatter plot of our data set is as shown below, can we guess the first principal component ? Yes, it's approximately the line that matches the purple marks because it goes through the origin and it's the line in which the projection of the points (red dots) is the most spread out. Or mathematically speaking, it's the line that maximizes the variance (the average of the squared distances from the projected points (red dots) to the origin).



The second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.

This continues until a total of p principal components have been calculated, equal to the original number of variables.

STEPS FOR RECOGNITION USING PCA

The step by step instructions along with the formulas for the recognition of faces using Principal Component Analysis (PCA) are as follows:

Step 1: Standardization

The aim of this step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.

More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.

Mathematically, this can be done by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standard\ deviation}$$

Figure 3 : Mean normalisation

Once the standardization is done, all the variables will be transformed to the same range [0,1].

Step 2: Covariance Matrix computation

The aim of this step is to understand how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

The covariance matrix is a p × p symmetric matrix (where p is the number of dimensions) that has as entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional data set with 3 variables x, y, and z, the covariance matrix is a 3×3 matrix of this from:

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Figure 4 : Covariance matrix for 3-dimensional data

Since the covariance of a variable with itself is its variance (Cov(a,a)=Var(a)), in the main diagonal (Top left to bottom right) we actually have the variances of each initial variable. And since the covariance is commutative (Cov(a,b)=Cov(b,a)), the entries of the covariance matrix are symmetric with respect to the main diagonal, which means that the upper and the lower triangular portions are equal.

Step 3: Compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components

Eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.

By ranking your eigenvectors in order of their eigenvalues, highest to lowest, you get the principal components in order of significance.

If we rank the eigenvalues in descending order, we get $\lambda 1 > \lambda 2$, which means that the eigenvector that corresponds to the first principal component (PC1) is v1 and the one that corresponds to the second component (PC2) is v2.

Step 4: Feature vector

Feature vector is simply a matrix that has as columns the eigenvectors of the components that we decide to keep. This makes it the first step towards dimensionality reduction, because if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions.

Last step :

Recast the data along the principal components axes

In this step, which is the last one, the aim is to use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components (hence the name Principal Components Analysis). This can be done by multiplying the transpose of the original data set by the transpose of the feature vector.

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

ALGORITHM

1. Read training images
2. Form training data matrix (M_{train_data})
3. Form training class labels matrix (M_{train_labels})
4. Calculate PCA transformation matrix (tmatrix)
5. Calculate feature vectors of all training images using tmatrix
6. Store training feature vectors in a matrix
7. Read test faces
8. For each test face do
9. Calculate the feature vector of a test face using tmatrix
10. Compute the distances between test feature vector and all training vectors
11. Store the distances together with the training class labels
12. Initialize error count to zero.
13. For each test face do
14. Using the distance data, determine the person ID of the most similar training vector
15. If the found ID is not equal to the ID of the test image increment error count
16. Output the correct recognition accuracy :

    (1 - (error count/ total test image count))*100

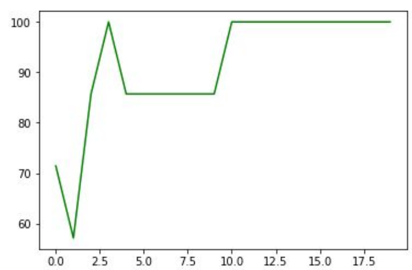
Results

With Probes dataset

```
In [174]: print("maximum accuracy = " , max(acc), "\nk = " , acc.index(max(acc)))

maximum accuracy =  100.0
k =  3
```

K vs Accuracy

```
In [175]: fig, ax = plt.subplots()

ax.plot(np.arange(20), acc, 'g')

Out[175]: [<matplotlib.lines.Line2D at 0x7fdb336dabe0>]
```



With Attachments dataset

```
In [80]: print(len(Sign_values) , len(Eigen_faces))

acc = []
for j in range(0,106):
    count =0
    for i in range(len(Test)):
        expected_index = Match_image(i,Sign_values[j],Mean,Eigen_faces[j])
        count +=  Compare(Train_img[expected_index],Test_img[i])
    acc.append(count/len(Test) * 100)
```

```
Bill_Gates_0005.jpg == David_Beckham_0011.jpg
Ariel_Sharon_0010.jpg == Arnold_Schwarzenegger_0007.jpg
Bill_Clinton_0007.jpg == Bill_Clinton_0010.jpg
Ariel_Sharon_0003.jpg == Anna_Kournikova_0012.jpg
Yoriko_Kawaguchi_0005.jpg == Arnold_Schwarzenegger_0005.jpg
Colin_Powell_0009.jpg == Ariel_Sharon_0011.jpg
Alvaro_Uribe_0004.jpg == Yoriko_Kawaguchi_0001.jpg
Alvaro_Uribe_0008.jpg == Bill_Gates_0009.jpg
David_Beckham_0009.jpg == Winona_Ryder_0008.jpg
Alvaro_Uribe_0008.jpg == Angelina_Jolie_0014.jpg
Alejandro_Toledo_0008.jpg == Arnold_Schwarzenegger_0011.jpg
Alvaro_Uribe_0010.jpg == Bill_Clinton_0004.jpg
Winona_Ryder_0001.jpg == Yoriko_Kawaguchi_0013.jpg
Alejandro_Toledo_0001.jpg == Alejandro_Toledo_0012.jpg
Andre_Agassi_0009.jpg == Anna_Kournikova_0007.jpg
Ariel_Sharon_0010.jpg == Bill_Clinton_0011.jpg
Andre_Agassi_0008.jpg == Bill_Simon_0005.jpg
Angelina_Jolie_0011.jpg == Abdullah_Gul_0008.jpg
Amelie_Mauresmo_0002.jpg == Alejandro_Toledo_0010.jpg
Yoriko_Kawaguchi_0005.jpg == Alvaro_Uribe_0005.jpg
```

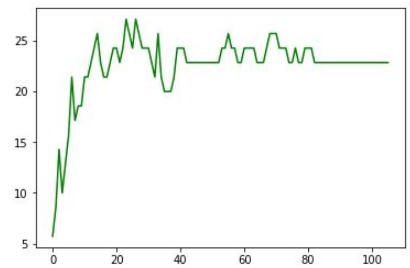
In [81]:
```
print("maximum accuracy = " , max(acc), "\nk = " , acc.index(max(acc)))
```

```
maximum accuracy =  27.142857142857142
k =  23
```

K vs Accuracy

In [82]:
```
fig, ax = plt.subplots()

ax.plot(np.arange(106), acc, 'g')
```

Out[82]:
```
[<matplotlib.lines.Line2D at 0x7fdb33db67f0>]
```



References

1. **[Steven M. Holland, Univ. of Georgia]**: Principal Components Analysis
2. **[skymind.ai]**: Eigenvectors, Eigenvalues, PCA, Covariance and Entropy
3. **https://towardsdatascience.com/a-step-by-step-explanation-of-principal-component-analysis-b836fb9c97e2**
4. **[Lindsay I. Smith]** : A tutorial on Principal Component Analysis
5. Face Recognition by Independent Component Analysis **Marian Stewart Bartlett**,Member, IEEE, **Javier R. Movellan**, Member, IEEE, and **Terrence J. Sejnowski**, Fellow,IEEE
6. Eigenspace-Based Face Recognition: A Comparative Study of Different Approaches **Javier Ruiz-del-Solar**, Member, IEEE, and **Pablo Navarrete**, Student Member, IEEE
7. www.wikipedia.com