IMAGE PROCESSING, RETRIEVAL AND ANALYSIS II: PROJECTS REPORT

Ahmad Alzeitoun

Institute for Informatik
University of Bonn,Bonn, Germany
Summer Term 2017

ABSTRACT

To achieve optimization in image analysis various advance methods will not only be discussed but their quality, performance, complexity, implementation simplicity will be discussed in this report. Possible usage of each technique along with advantages of a particular method w.r.t application over one another will also be explained. No doubt, computer vision is very demanding and challenging field now a days which help to solve image analysis problem in various techniques to achieve certain goals.But in this report we will cover only few techniques ranging from image intensities to tensor linear discriminant analysis. Gray scale image analysis will be discussed in detail using computer vision procedures as computer vision is super set of image analysis. Flow of report will move on from image quantization to wards its recognition and give meaning to its features. Loly-Max algorithm is used to quantize image as well as illumination compensation techniques to illuminate it, therefore, Principal Component Analysis (PCA) using single value decomposition is adopted for eigenfaces. Morever,Fisher's Linear Discriminant Analysis(LDA) has been used for image classification which leads to demonstrate Tensor Linear Discriminant Analysis(TLDA) for optimization and to mitigate all issues faced in conventional LDA. In last but not least these approaches has been compared term of performance, quality, computational speed and efficiency and not only explained with theory only but also with practical implementation to demonstrate the results which shows excellent agreement with theory and analytical results.

***Index Terms*—** image intensities,quantization,Lloyd-Max algorithm,level, histogram,numerical robustness, Dirty Trick,Eigenfaces,Single value decomposition (SVD),Principal Component Analysis(PCA),Classification, Linear Discriminant Analysis (LDA),Least Square Regression(LSE),Eigenvalue and vector, Tensor LDA

1. INTRODUCTION

Extraction of information which is meaningful from images using various methods is called image analysis. [20]. It is subset of computer vision and help to build artificial systems that extract useful information from various sources like images,video etc. [2, 15, 17, 20, 21, 23]

It is also called image understanding and is at mid to higher level process which involve image segmentation(dividing an image into parts,object or regions, their description and classification of individual objects where mostly images are input and output is extraction of those attributes from input images as well as making sense of an ensemble of recognize objects. [2, 10, 12, 15, 17, 20, 21, 23]

It characterize a shap, structure and constituents connectivity quantitatively and is important for various applications and science and technology. [15]

Therefore, scope of this report is only "Image Processing, Retrieval and Analysis II" lectures which are limited to intensity transformations, principal component analysis,face recognition, face detection, template matching,image features, local descriptors etc along with performance advantages of each justified with not only theory but also with practical implications. [2]

Moreover, all 3 projects which involve image intensities (quantizaiton and illumination), eigenfaces using PCA and binary classification using conventional LDA and (tensor)Linear Discriminant Analysis(LDA) will be demonstrated in this report. Therefore for implementation c/c++, OpenCV,Gnuplot and linux operating system are used.Therefore report structure involves problem definitions, purpose, theoretical background required to achieve the desired goals,results achieved, performance measures of one method over another,benefits, efficiency, computational speed, error measure code quality and conclusion.

Furthermore, image analysis tasks may be simple or sophisticated at the same time like reading a bar code or identifying an objects from millions of objects.

2. THE PURPOSE OF THE PROJECTS

Mastering advance image analysis techniques required to implement computer vision algorithms by comparing various methodologies w.r.t complexity, performance optimization ,code quality and image quality to solve a single problem in various ways on the basis of results obtained is the goal of these projects. In short it provide a better experience of turning facts into reality. It helps not only to study the theory but also practical experience by implementing these techniques for deep understanding with real time results at visual level.

This goal is well summarized by the saying "A picture is worth a thousand words but a word can change its meaning".

3. IMAGE INTENSITIES

Grey value quantization and illumination compensation problems are solved and analysis of result's visual quality, implementation simplicity and performance of method adopted has been discussed in this section.

3.1. Quantization

Transformation or mapping of continuous to limited number of discrete gray values is called quantization. [2, 12] it compress the range of values to single quantum value.e.g reduction in number of colors reduces image file size [2, 12, 20]

Therefore this compression is lossy.It constrain something from continuous to a discrete. Basically it divide an image into quanta(partitions) which is opposit of sampling and it digitizes amplitudes.

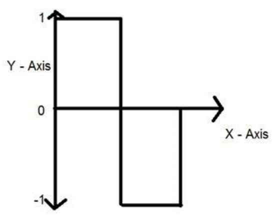


Fig. 1. quantization [23]

In figure 3.1 , a signal has been quantized into 3 levels and values are set to these to these levels.There are at max $256 = 2^8$ intensity levels per channel to store images with discrete values [2]

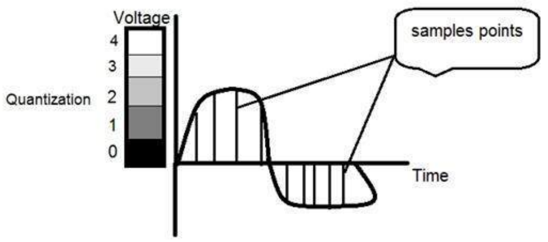


Fig. 2. quantization levels [23]

In the figure 3.1 vertical ranging values are quantized into 5 different levels and resulting image have 5 different black and white or gray colors.Therefore, increasing number of levels improves image quality both are directly proportional to each other. In this report gray scale image at L=256 is choosen and each level is called gray level. [12, 23]

Gray level = number of levels per pixel=$L = 2^k$.

True value replaced by quantization level leads to an error and if the levels are equally spaced with distance and all gray values are equally probable, then variance introduced by quantization is as under:

$$\sigma_q^2 = \frac{1}{\Delta g}\int_{g_q-\Delta g/2}^{g_q+\Delta g/2}(g-g_q)^2 dg = \frac{1}{12}(\Delta g)^2 \tag{1}$$

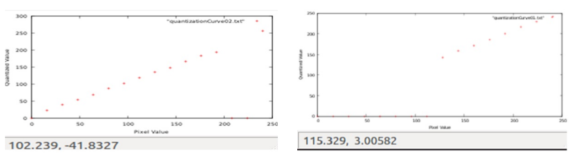
The level g_q for which the distance from the gray value g, $|g - g_q|$, is smaller than the neighboring quantization levels.The standard deviation σ_q is about 0.3 times the distance between the quantization levels. Mean error is approximately 0.3 times quantization levels and max error is half of quantization level for equally spaced quantization.Therefore,mean value error decreases with the number of measurements.

$$\sigma_{mean} \approx \frac{1}{\sqrt{N}\sigma} \tag{2}$$

where σ = standard deviation of individual measurement N =Number of measurement taken. For repeated measurements without noise the quantized value will be same otherwise different. [12] Therefore modern digital communication systems require more sophisticated quantization techniques. So Lloyd-Max Algorithm for gray value quantization has been implemented.

(a) Original (b) Quantized Dark Image (c) Original (d) Quantized Image

Fig. 3. Gray Value Quantization at Level 16 by Lloyd-Max



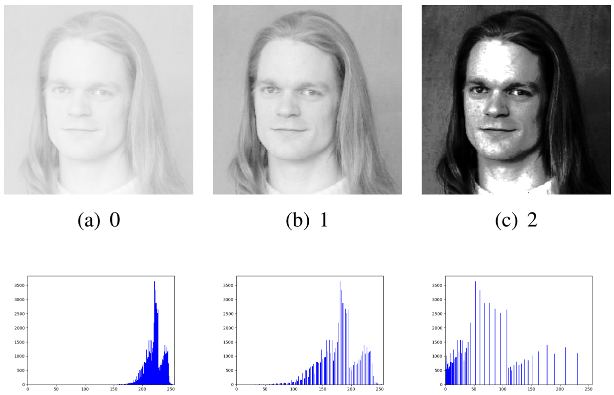
(a) Dark Image (b) Bright Image

Fig. 4. Quantization Curve for dark and relatively bright image

Since Lloyd-Max Algorithm is based on K-Means and is optimal for fix number of levels. but levels may also be generated dynamically by using growing gas or some other techniques to make it optimal.Moreover,low number of gray values leads to false edges and makes it very difficult to recognize objects that show slow spatial variation in gray values. [2, 12, 17]

3.2. Histogram Transformations

The core idea in histogram transformation is applying normalizing operation on the original image. By drawing normalized histogram we can note that the lines are distributed and this distribution represent change in pixel intensity values. we have to find exponential function then apply equalization and draw equalized histogram, we will note the lines are distributed more and that will reflect more changes in intensities of the pixels in the image. The figure5 illustrate the changes in intensities and how the lines are distributed.



(a) 0 (b) 1 (c) 2

(d) 3 (e) 4 (f) 5

Fig. 5. Histogram Transformation

3.3. Illumination Compensation

Illumination is the process of equalizing the light on scene image.An illumination is a function which assigns intensity values to certain image. and an image is result of illumination and reflecting of a recorded scene. [**?**, 2, 17] Therefore,images with in homogeneous illumination are used here for testing

$$f(x, y) = i(x, y) \cdot r(x, y) \tag{3}$$

where $i = illumination$ and $r = reflectance$and $f(x, y)$ is an image pixel value at point x and y
when there is a light on scene object it not only illuminate but also reflect and what we see is basically the combination of illumination and reflectance. Furthermore,illumination depends upon few light sources and is a function that vary slowly while reflectance depends upon physical properties of the object and different object have different properties.If illumination is known reflectance can be solved. [2]

$$r(x, y) = \frac{f(x, y)}{i(x, y)} \tag{4}$$

Lets try to determine illumination by mean of model fitting and if it vary linearly then a plan may be fit into it. For numerical robustness and to avoid overflow and underflow following function is considered.

$$l(x, y) = \ln f(x, y) \tag{5}$$

In practice, when image is read 1 is added to every pixel for numerical robustness but need to keep the number small to do it in log space is called **Dirty Trick** [2].

$$\begin{aligned} \text{l(x,y)} &= \ln\left(f(x, y) + 1\right) \qquad\qquad "DirtyTrick" \\ &= \ln\left(i(x, y)\right) + \ln\left(r(x, y)\right(\\ &= i_l(x, y) + r_l(x, y) \end{aligned}$$

Assume a certain model for $i_l(x, y)$ and fit it to $l(x, y)$ and compute reflectance as under

$$r_l(x, y) = l(x, y) - i_l(x, y) \tag{6}$$

and

$$r(x, y) = e^{r_l(x,y)} \tag{7}$$

3.3.1. *More Dirty Tricks*

First rescale intensities in $r(x, y)$ to range in $f(x, y)$,i.e

$$r \leftarrow \frac{r - r_{min}}{r_{max} - r_{min}} \cdot (f_{max} - f_{min}) \tag{8}$$

the new image may look darker than the original image so adjust mean of $r(x, y)$ to mean of $f(x, y)$,i.e.

$$r \leftarrow r - \mathbb{E}[r] + \mathbb{E}[f] \tag{9}$$

Basically average brightness of the new image is subtracted from the image and average brightness of the image is added to it i.e equalize $r(x, y)$

3.3.2. *Model Fitting*

For experiments a linear model is fitted into illumination as below.

3.3.3. *Fitting a Linear Model*

Image is a combination of illumination and reflectance [2] and it is truned into something that better fitt into subsequent analysis. **Ansatz:**$i_l(x, y) = ax + by + c$
and our **goal** is to minimize the error E

$$min_{a,b\ ,c} = \sum_{k=1}^{N} (ax_k + by_k + c - l(x_k, y_k))^2 \tag{10}$$

It is $3D$ regression plane fitting problem and data sets are labeled as

$$\{(x_k, z_k t)\}_{k=1}^{N} \tag{11}$$

where $x_k = \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix} \in \mathbb{R}^3$, $z_k \in \mathbb{R}$ if $w = \begin{bmatrix} a \\ b \\ c \end{bmatrix} \in \mathbb{R}^3$ the equation 10 becomes as

$$min_w E = \sum_{k=1}^{N} (x_k^T w - z_k)^2 = \|Xw - z\|^2 \tag{12}$$

where $X = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} \in \mathbb{R}^{N\times 3}$, $Z = \begin{bmatrix} z_1 \\ \vdots \\ z_N \end{bmatrix} \in \mathbb{R}^N$ therefore

$$\|Xw - z\|^2 = w^T X^T X w - 2wTX^T z + z^T z \tag{13}$$

by partial derivation

$$w = (X^T X)^{-1} X^T z \tag{14}$$

Following procedure 1 has been adopted for implementation of linear illumination compensation model.

Algorithm 1 Linear Model Illumination Compensation [2]

LM Illumination Compensation Compute the logarithm for each cell Create matrix of pixels in the following format: $x = \begin{bmatrix} x_1^T \\ \vdots \\ x_N^T \end{bmatrix} \in \mathcal{R}^{N*3}$ where $x_k = \begin{bmatrix} x_k \\ y_k \\ 1 \end{bmatrix}$

Create matrix with pixel values in a form of a vector Calculate linear model coefficient using equation:

$$W = (X^T X)^{-1} X^T Z \text{ where } w = \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

Having coefficient of linear model: Compute Illumination Plane i.e $ax + by + c = z$ Compute Reflectance i.e $r(x, y) = l(x, y) - i(x, y)$

(a) Original (b) Reflectance (c) Linear Illumination Compensation Model

Fig. 6. Linear Model Illumination Compensation

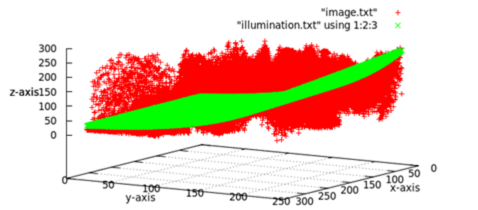


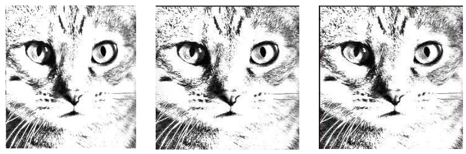
Fig. 7. 3D graph of the image function f(x,y) together with the resulting illumination model $i(x, y)$

The full image linear model illumination compensation can be seen in the figure 3.3.3 where reflectance and its linear model illumination has be demonstrated as well as in 3.3.3 it can be seen clearly.

(a) Original (b) reflectance (c) illumination

Fig. 8. Full Image Linear Model Illumination Compensation

Now lets have a look at some other random 100, 250 and 1000 samples and compare the mean square error.



(a) 100 (b) 250 (c) 1000

Fig. 9. Random Samples Reflectance

(a) 100 (b) 250 (c) 1000

Fig. 10. Random Samples Illumination

Mean Square Error

A	B	C	Error
100 Samples			
0.0129936	-0.0014632	2.6863863	2.38126e+08
250 Samples			
0.0121953	-0.0017362	2.9454632	2.35856e+08
1000 Samples			
0.0122765	-0.0011592	2.7700524	2.23717e+08
Whole Image			
0.0112255	-0.0009097	2.903678	2.09369e+08

It is clearly seen that increase in number of samples, decreases the mean square error which is the goal of illuminaiton compensation to minimize the error as much as possible.Thus the more samples are taken into calculation the better linear model is obtained [2, 10, 12, 15, 17, 20, 21, 23]

4. EIGENFACES

This chapter demonstrate how to compute eigenfaces. A facial image is a point from a high-dimensional image space. The PCA help to found the lower-dimensional subspace and identifies maximum variance axes which results in easy classification. This is optimal transformation from a reconstruction standpoint and doesnt take any class labels into account. The maximum variance axes do not necessarily contain any discriminative information. Within class vairance need to be minimize while maximizing it between the classes at the same time. [2, 4, 6, 7, 9, 11, 15]

Basically Eigenfaces approach capture the variations in a collection set of face images and use this info to compare and encode individual faces in a holistic manner. They are eigenvectors of the covariance matrix of face images and principal components of a distribution of faces. It is considered as first face recognition tech and served as basis of commercial face recognition tech products. It has many new extensions to original method and developments in automatic face recognition products. Its initial idea developed by Sirovich and Kirby 1987 while Turk and Pentland used it in 1991 for face detection and recognition. [**?**, 1, 2, 7]

Moreover,to minimize complexity each face image can be represented using a small number of parameters. Eigenfaces are set of features and among face images these features characterize the global variation.Then each face image use a subset of eigenfaces approximately [2, 11]

The mapping of high dimensional data to lower dimensional space to filter out noise and extract latend info from data is the goal of dimensionality reduction.Therefore, for dimensionality reduction singular value decomposition is commonly used.

4.1. Single Value Decomposition (SVD)

SVD is a powerful linear algebra factorization technique of a rectangular real or complex matrix; it works even for singular or numerically near-singular matrices. It is used with many applications for solving linear equations.

4.1.1. SVD Theorm

Every matrix $X \in \mathbb{R}^m$ can be factorized as $X = U\Sigma V^T$ where

$U \in \mathbb{R}^{m\times m} \Leftrightarrow$ orthognal matrix of **left singular vectors**
$V \in \mathbb{R}^{n\times n} \Leftrightarrow$ orthognal matrix of **right singular vectors**
$\Sigma \in \mathbb{R}^{m\times m} \Leftrightarrow$ diagonal matrix of **singular values**

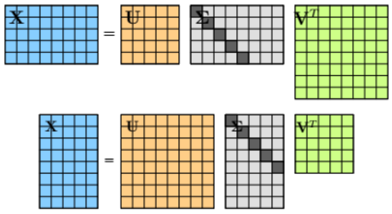


Fig. 11. SVD Theorm [2, 17]

Now using SVD

$$min_{rk(X_k)=k} \|X - X_k\|^2 = \sum_{j=k+1}^{r} \sigma_j^2$$

where $X_k = U_k \Sigma_k V_k^T = \sum_{j=1}^{T} \sigma_j \mu_j \nu_j^T$

4.2. Principal Component Analysis (PCA)

It is also called (discrete) **Karhunen-Loeve Transfor (KLT)** or **Hotelling Transform**.It convert high dimensional data set to lower dimension.It represent data in new coordinate system and divides observed space into orthogonal subspaces with maximum variance. It can be used for **lossy data compression**by retaining those characteristics of datset which contribure most to its variance. It transform a number of possibly co-related variable into uncorrelated variables known as **principal components**. [1, 2, 7, 11, 15, 17, 21, 22]

Let a dataset consist of N observations, each of M variables(dimensions).where $N >> M$. The intention is to reducse the dimensionality of data so that each observation can be usefully represented with only L vaiables, where $1 \leqslant L < M$. Data are arranged as a set of N column data vectors, each representing a single observation of M variables: the n-th observations is a column vector $X_n = (X_1 \cdots X_M)^T$ where $n = 1 \cdots N$. So we have M x N data matrix R. It is very larg matrics as N may be very larg.This procedure is applied to normalized data X as follows. The raw observed data is arranged in a matrix and the average (empirical) mean along each row of matrix is calculated as under.The result is stored in a vector μ the element of which are scalors: [2, 17]

$$\mu(m) = \frac{1}{N}\sum_{n=1}^{N} R(m,n) \text{where } m = 1 \cdots M \tag{15}$$

The average mean is sbutracted from each column of R: if e is a unit vector of size N then $X = R - \mu e$. If we approximate X in a lower dimensional space M by the lower dimensional matrix Y (of dimension L) , then the mean square error ϵ^2 of this approximation is as under:

$$\epsilon^2 = \frac{1}{N}\sum_{n=1}^{N} \|X_n\|^2 - \sum_{i=1}^{L} b_i^T \left(\frac{1}{N}\sum_{n=1}^{N} X_n X_n^T b_i\right) \tag{16}$$

where $b_i\,(i = 1 \cdots L)$are basis vector of the linear space of dimension L. Therefore to minimize the MSE(ϵ^2) then following term has to be maximized.

$$\sum_{i=1}^{L} b_i^T C b_i \tag{17}$$

where

$$C = \sum_{n=1}^{N} X_n X_n^T$$

is the Covariance matrix.

The covariance matrix C has special properties: it is real, symmetric and positive semi-definite. In the data approximation,dimensions corresponding to the smallest eigen-values are omitted. The mean square error ϵ^2 of equation 16 becomes as under:
$\epsilon^2 = trace(C) - \sum_{i=1}^{L} \lambda_i = \sum_{i=L+1}^{M} \lambda_i$
where $trace(C)$ is the sum of the diagonal elements of a the matrix and its equal to sum the sum of all eigen values. [17]

4.3. Computing Eigenfaces

90% of images i.e 2186 has been chosen randomly as training images from a collection of 2429 tiny face images of size 19×19 and remaining 10%i.e 243 as test images. The images are read into a matrices $X_{train}^{361x2186}$ and $X_{test}^{361x243}$ respectively and data is centered to zero mean first. The adopted procedure for PCA and computing Eigen Faces is elaborated in algorithm 2 and 3 respectively. Since C is positive semidefinite and

---

Algorithm 2 Principal Component Analysis [2]

Computing Eigen Faces given: data $\{z_i\}_{i=1}^{n} \subset\in \mathbb{R}^m$, $\mu = \frac{1}{n}\sum_i Z_i$ transform to zero mean $x_i = z_i - \mu$ and obtain matrix $X = [x_{1n}] \in \mathbb{R}^{m \times n}$ compute data covariance matrix i.e.$C = \frac{1}{n}\sum_i x_i x_i^T = \frac{1}{n} X X^T \in \mathbb{R}^m$ determine all solutions to:$Cv_j = \lambda_j v_j$ obtain matrix: $V = [v_1 \cdots v_m] \in \mathbb{R}^m$ and compute:$V^T X$

---

symmetric then it has the following properties:1.Eigenvalues λ_j are real.2.eigenvectors $v_j$and$v_l$are orthogonal if$\lambda_j \neq \lambda_l$ 3. **C** is not defective, it has $m$independent eigenvectors 4. $V = [v_1 v_{2m}] \in O(m) \Rightarrow VV^T = V^{-1}$ 5.V diagonalizes C,i.e. $C = \sum_j \lambda_j v_j v_j^T$

Algorithm 3 Computing Eigen Faces

Computing Eigen Faces Load 90 percent random images into train matrix Load remaining 10 percent into test matrix Calculate mean of train matrix Centered train images by subtracting mean Calculate co-variance matrix by outer products Calculate eigen vectors and eigenvalues Find minimum k Get test image and project it into face space.

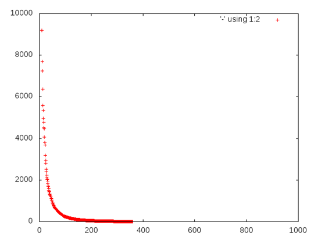


Fig. 12. Spectrum of C (Eigen Value Graph)

decending order eeigenvalues as showed in above graph and there is no change anymore in the eigen values after index 200 or 250. Eigen values leads to decision that how many eigen-faces need to be used.The small value eigen-vectors have info about detailed differences in w.r.t to large value eigen-vector at higher level discrimination.Therefore vectors with smaller eigenvalues may be removed without affecting the results which leads to descending order sort of eigenvalues as shown in graph.

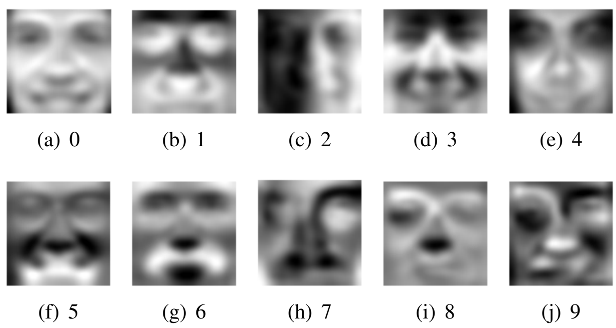
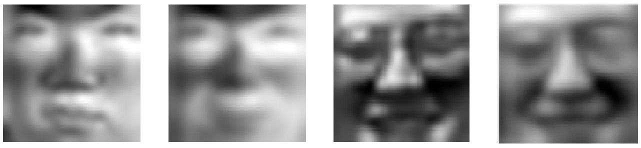


Fig. 13. First k Eigen Vectors

Nearest Neighbour for Images	
Image	**Identical**
0	yes
1	yes
2	no
3	yes
4	yes
5	yes
6	yes
7	no
8	yes
9	no



(a) Original Space (b) Low Dimensional Space (c) Original Space (d) Low Dimensional Space

Fig. 14. Projection

So number of eigenvectors reduced from 361 to 19 and its a big compression therefore images in lower dimensional space are quite similar to original space. There is not much difference between distances in these two spaces. In low dimensional space slightly lower.Furthermore, all nearest neighbors are not identical as shown in table above.

5. BINARY CLASSIFICATION

In this report image has been quantized, illuminated and eigenfaces has been computed using different methods.Now objects are localized in an image by giving meaning to them in this section as classification is the problem of assigning lables .i.e. giving meaning to data. Assume certain kind of object need to detect from an image therefore classifier need to train to distinguish required objects. [2, 3, 5]

As PCA describe objects but does not distinguish between classes and finds accurate data representation in a lower dimensional space by projecting data in the direction of maximum variance which may be useless for classification.Since Fisher Linear discriminant project data to a line which maintain direction useful for data classification which solves the problem of classification [2] i.e to detemine a suitable projection vector w from the training data and choose the line that maximally separate the projected data. [2, 3, 5, 15]

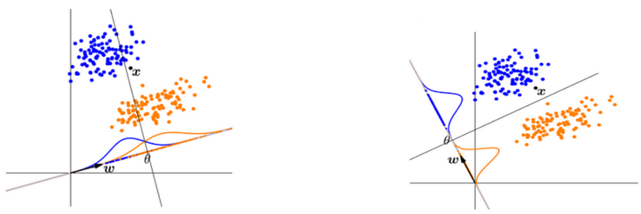
5.1. Fisher's LDA for Binary Classification

A linear function $w^T x$ that maximizes the following objective

$$J(w) = \frac{(\mu_1 - \mu_2)^2}{s_1^2 + s_2^2} \tag{18}$$

is called Fisher's Linear Discriminant [**?**, 2, 5, 17, 18]. where
μ_1 = Projected mean of class1
μ_2 = Projected mean of class2
s_1^2 = samples of classe1 cluster aroundμ_1
s_1^2 = samples of class2 cluster aroundμ_2
$\mu_k = \frac{1}{n_k}\sum_{x\in\Omega_k} x$ = sample class mean
$\mu_k = \frac{1}{n_k}\sum_{y\in\Omega_k} y = w^T \mu_k$ = Projected sample class mean
$s_k^2 = \sum_{y\in\Omega_k}(y-\mu_k)^2$ = Projected class scatter

Note*: maximizing $J(w)$ w.r.t. the training data will identify a projection direction w such that samples from either class are projected close to each other and the projected class means are as far separated as possible.
Therefore **goal** is to maximize the separation of means as far as possible and minimize the variance(scatter in classes) as much as possible.



(a) Bad example [2] (Means are not separated) (b) good example [2](means are well separated)

By expressing J(w) explicitly a function of$w$to find optimal w^* $s_k^2 = w^T s_k w$
$s_1^2 + s_2^2 = s^T S_w w$
$(\mu_1 - \mu_2)^2 = w^T S_B w$
where B is between class scatter matix
so equation 18 becomes as

$$J(w) = \frac{w^T S_B w}{w^T S_w w} \text{Rayleigh quotient} \tag{19}$$

Therefore to maximize it with respect to w , consider its derivation and therefore 19 becomes

$$J(w) = s_w^{-1} S_B w \tag{20}$$

Equation 19is generalized eigenvalue/eigen vector problem, so therefore by solving it

$$w^* = argmax_w \frac{w^T S_B w}{w^T S_w w} = S_w^{-1}(\mu_1 - \mu_2) \tag{21}$$

Here a class of cars is needed to distinguish from every thing else therefore two sets collection of images, namely +ve and -ve examples are given which includes 2556 training image patches of which 2442 are background, while others 124 contain cars and their size is 81×31.This binary problem is solved using LDA and The main idea of Fisher's LDA is to find projection to a line s.t samples from different classes are well separated.

Algorithm 4 LDA Algorithm [2]

LDA Select random number from range between projection means Create 10 different classifiers
$\theta_k \in [\mu_1, \mu_2] k = [1 \cdots 10]$ and

$$y(x) = \begin{cases} +1, if w^T x \geqslant \theta_k \\ -1, o/w \end{cases}$$

Evaluate classifier by projecting data on w vector and calculating precision and recall $w^t x \geqslant \theta_k$ Calculate mean: $\mu_k = \frac{1}{n_k} \sum_{x \in \Omega_k} x$ Caculate co-variance matrix

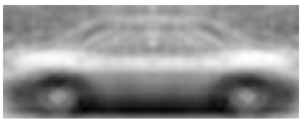
$$s_k = \sum_{x \in \omega_k} (x - \mu_k)(x - \mu_k)^T$$

Sum co-variance for negative and positive Calculate weight matrix

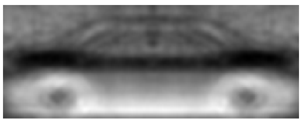
$$w* = argmax_w \frac{w^t S_\beta w}{w^T S_W w} = S_w^- 1(\mu_1 - \mu_2)$$

Project data on weight vector and calculate mean **//LDA Evaluation on Test Set** each test image co-relate image and weight matrix check if co-relation for each pixel is greater than threshold

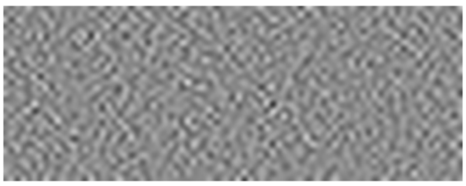
Let have a look at results after applying LDA algorithm 4 on giving data.



(c) +Ve Mean



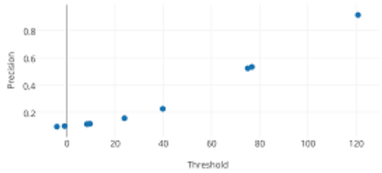
(d) -Ve Mean



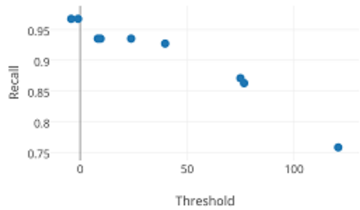
(e) Weight Matrix

Fig. 15. LDA Results: Means and Weight Matrix

In the above figure 15 cars can be easily identified in positive and negative mean but not in the case of weight matrix obtained, although in weight matrix means are far separated and variance is as small as possible. Weight matrix looks very noisy because it has many free parameters and very flexible to learn noise although this classifier separate training data very well but case is different in case of testing. Therefore figure 16 characterize the classifier performance by means of precision/recall curves.

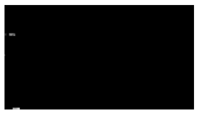


(a) LDA Precision

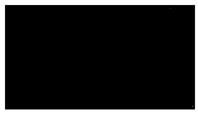


(b) LDA Recall

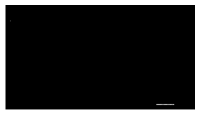
Fig. 16. LDA Precision and Recall



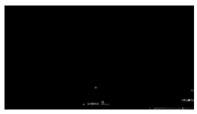
(a)



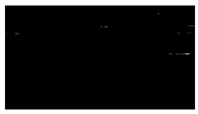
(b)



(c)



(d)



(e)

(f)

Fig. 17. LDA Car Detection

In figure 17 cars labels are not clearly visible its due to some issues with LDA.Therefore its EER is 60 % to 75%.

5.2. Problem with LDA

Assume that a sample $x_i{}_{i=1}^{n}$ where $x_i \in \mathbb{R}^m$ then s_w may be ill conditioned or even singular which mean s_w^{-1} may not exist. and this usually happens for $m >> n$ and is known as small sample size problem.Furthermore,for $m >> 1$, matrix $s_w \in \mathbb{R}^{m \times m}$ is huge which leads to long computation. [2]

5.3. Tensor LDA Binary Classification

LDA is well know for supervised dimension reduction.TLDA provides a good relationship with conventional LDA. [13]

Tesnor LDA not anly solves all issue that normal LDA faces. It improves complexity,quality and is much faster and efficient the conventional one. [2, 3, 5]

TLDA is more elaborative way of doing LDA and is variant of conventional LDA.Only fewer parameters are determined in it and it does not suffer from sample size and ill condition problem like LDA.It can also works with few training examples.It is separable which mean learned classifier are separable and are sort of vectors. Moreover, separable template/classifiers applied to certain image pixels leads to improve computational speed and this efficient execution is very desirable. [2, 3, 5, 8, 14, 16, 25]

Some time its beneficial to treat images as tensors(a three way array). They describe linear relations between geometric vectors, scalars, and other tensors [2, 26]e.g the dot/cross product or linear maps etc.

Moreover, a tensor can be represented as an organized multidimensional array of numerical values. Its order/degree is the dimensionality of the array needed to represent it. A vector represented as a 1-dimensional array is 1st-order tensor. Scalars are 0th-order.Therefore tensors themselves must be independent of a particular choice of coordinate system. [2, 26]

5.4. Mathematical Preliminaries

Here are some basic terminologies that used in algorithms for 2nd order tensor.
Inner Product

$\langle A, B\rangle = A_{ij}B_{ij}$ by using Einstein's summation convention

Rank-1 Tensor

$$A = UV^T = U \otimes V \text{ i.e } A_{ij} = U_i V_j$$

Therefore, if there is

$$A* = \sum_r U_r \otimes v_r \text{ s.t. } A* = argmin \left\| A - A' \right\|_F$$

then A has a PARAFAC Model and $\otimes$ denotes vector outer product [2, 3]. We know that

$$W = \sum_{r=1} U_r^T X V_r$$

$$X.W = U^T XV \text{ [2,3,5]}$$

Therefore, convolving an image I with W without using any constrain
$Y_{ij} = (I * W)_{ij}$. Its complexity is $O(mn)$ per pixel
By using PARAFAC constrain [2]

and the Y_{ij} is the projection of image patch X_{ij} centered at image coordinate(i, j) Moreover, tensor LDA using Alternating least squares (ALS) is applied as under where training sample $\left(X^l, y^l{}_{l=1}^{N}\right.$ is given and we have to determine $W = uv^T$. Therefore for training and in application phase data has been transferred to zero-mean for the sake of simplicity.Therefore procedure refalg:tlda is used for tensor LDA experiments with the same data as for conventional LDA.

Algorithm 5 Tensor LDA Algorithm [2, 3, 5]

---

tLDA Randomly initialize u and v Compute the contraction$x_j^l = u_i X_{ij}^l$,$x_i^l = X_i j^l v_j$

apply least square method

$$argmin_v ||Xv - y||^2,$$
$$\text{argmin}_u ||Xu - y||^2,$$
u=inv((XU.trans()*XU))*Y ,
V=inv((XV.trans()*XV))*Y

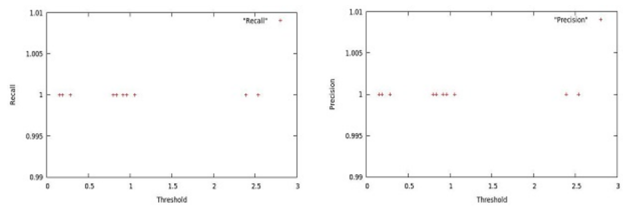
$(\mu \leqslant \epsilon)$ Orthogonalize U and V $W+ = U * V - trans$
//Classification Calculation$Select random number from range betw$
$[\mu_1, \mu_2]$ evaluate classifier by projecting data on w vector and calculating precision and recall $w^T x \geqslant \theta_k$

5.5. Results

(a) $\rho = 1$ (b) $\rho = 3$ (c) $\rho = 9$

Fig. 18. Tensor LDA Weight Matrix

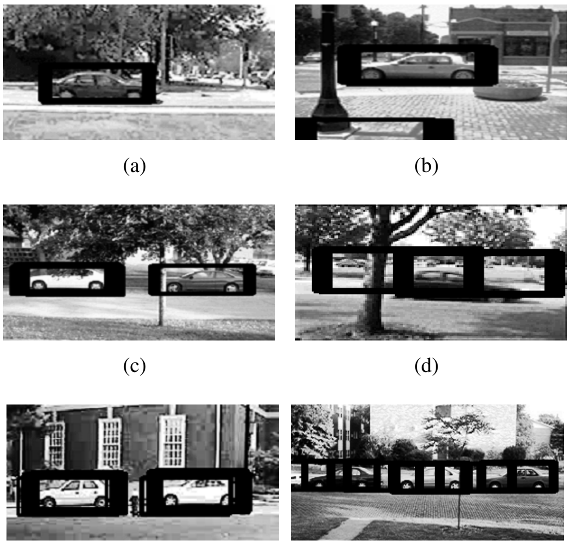
In figure 18 car like images in all 3weight matrix can be seen clearly because there are less number of parameters and classifiers are separable.



(a) Tensor LDA Recall (b) Tensor LDA Precision

Fig. 19. Tensor LDA Precession and Recall

Figure 19 characterizes classifier performance by means of precision/recall curves. Maximum precision and recall in the above precision and recall graphs can be seen easily.Both increases with the threshold. Precision and recall intersect the diagonal and at $\rho = 9$ the TLDA perform much better. In comparison to LDA it works better when testing it. LDA separate training data very well but what learned from training data does not really generalized to the application phase. Because the training data does not really represented it well.Let have a look now how accurately cars are labeled.



(a) (b) (c) (d) (e) (f)

Fig. 20. Tensor LDA Car Detection

In figure 20 (b and f) there are some mis-classifications.In f its due tyres as it classify tyres as car as it has learned that a car has tyres and a body therefore these mis-classifications are interpretable. TLDA yields not only best performance but also shortest training time and best run time.Its EER is more than87% [2, 3, 5]

Furthermore, it is also demonstrated that LDA takes more time to train while the Tensor LDA is relatively faster (takes less time) LDA is more susceptible to small changes in the data because number of parameters to learn are $(n * m)$ while Tensor LDA is not so susceptible to noise because number of parameters to learn are smaller i.e $n + m$. Moreover LDA does not work in practice if number of training examples are less than dimensionality of the data.

5.6. LDA vs Tensor LDA

LDA has many free parameters to be learned and is so flexible to learn noise.Basically it leans every tiny variation in back ground.It learns training data very well but practically,it does not do much with test data [1–3, 5–7, 13, 14, 16, 21, 22, 24, 25]

While tensor LDA is not efficient in separating training data although its faster. Actually by design its classifier is not so flexible with training data and does not have the luxury to learn noise.There are reduced number of parameters as classifier is contrained in TLDA and can be applied very huge template. It is much faster that conventional LDA because it has to train only fewer parameters and is separable.Moreover, TLDA does not suffer from small sample size or ill conditioned problem .It is robust against noise and over fitting.

In conventional LDA scatter matrix may be singular because the number of training examples are much smaller than its dimensionality while required metrics are smaller in case of tensor LDA which does not suffer form such problems.Therefore the equal error rate (EER) in case of LDA is less than 70% and in case of TLDA it greater than 87% [1–3, 5–7, 13, 14, 16, 18, 19, 21, 22, 24, 25]

6. CONCLUSION

Image analysis is a complex field, as it heavily relies on mathematics, Machine learning, data mining, optics and computer science. The techniques and methods explained in this report are good approaches to achieve the results.Therefore, there may be many other methods to achieve the same goals.In this report different flavors of quantization,illumination has been seen practically along with error rate. Eigen faces computation is also demonstrated using eigenvalue and vector along with PCA.Both conventional LDA and tensor LDA not only implemented but are also compared in term of performance, quality, efficiency and complexity.

7. ACKNOWLEDGMENT

First of All I am very thankful to Allah subhan wa tallah, without his blessings I can never do anything then ever support of my family. I am also very thank full to my team who share the ideas and we work and cofee together. The most important, I am extemely thankful to **Prof.Dr.Christian Bauckhage**, without his guidance, advices, lectures, support and ever helping, kind and friendly attitude we might not be able to explore the field in a optimal angel.He taught us how to turn the possibilities into facts and get optimal solutions not only in the field of Image Analysis and computer vision but also every field of life too.

8. REFERENCES

[1] Matthew A.Turk and Alex P.Pentland. Face recognition using eigenfaces. *IEEE*, CH2983-5/91-00/0586:586–591, 1991.

[2] Christian Bauckhage. Image processing, retrieval and analysis ii (lecture notes), April 2015.

[3] Christian Bauckhage and John K. Tsotsos. Separable linear discriminant classification. *W. Kropatsch, R. Sablatnig, and A. Hanbury (Eds.): DAGM 2005, LNCS 3663, pp. 318325, 2005. Springer-Verlag Berlin Heidelberg 2005*, 2005.

[4] Morelock Bursn. A comparison of face recognition methods. its a Final Project Report, 2003.

[5] Thomas Kaster Christian Bauckage. Benefits of separable, mulitilinear discriminant classification. *IEEE*, 2006.

[6] OpenCV Documentation. *Face Recognition with OpenCV*. OpenCV, 2.4.1.1.0 edition.

[7] Dr. Matthew Turk Dr. Sheng Zhang. Eigenfaces. *Scholarpedia*.

[8] Kamran Etemad and Rama Chellappa. Discriminant analysis for recognition of human face images. *Optical Society of America*, 1997.

[9] Daniel Georgescu. A real-time face recognition system using eigenfaces. *Journal of Mobile, Embedded and Distributed Systems*, 3(4), 2011.

[10] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[11] Ankit Gupta. Eigenfaces.

[12] B. Jahne. *Digital Image Processing*. Springer, 2005.

[13] Lei Shi a Shulan Li c Jianhua Zhao a, Philip L.H. Yu b. Separable linear discriminant analysis. *Computational Statistics and Data Analysis*, 2012.

[14] William Kunz Jie Yang, Hua Yu. An efficient lda algorithm for face recognition.

[15] H. De Raedt K. Michielsen. Morphological image analysis. *Computer Physics Communications 132 (2000) 94103*, 2000.

[16] N. Kehtarnavaz and M.F. (eds.) Carlsohn. Two dimensional statistical linear discriminant analysis for real-time robust vehicle type recognition. *SPIE*, 2007.

[17] Roger Boyle Milan Sonka, Vaclav Hlavac. *Image Processing, Analysis, and Machine Vision(3rd edition))*. Thomson Learning.

[18] Simon J. D. Prince. *Computer Vision: Models, Learning, and Inference*. Cambridge University Press, New York, NY, USA, 1st edition, 2012.

[19] David G. Richard o. Peter E. *Pattern Classification (2nd Edition)*.

[20] Breckon Solomon, C.J. *Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab.* Wiley-Blackwell, 2010.

[21] Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010.

[22] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 1991.

[23] TutorialsPoint. Digital image processing tutorial.

[24] Martina Uray. *Incremental, Robust, and Efficient Linear Discriminant Analysis Learning*. Phd theis, Graz University of Technology Institute for Computer Graphics and Vision, Austeria, 2008.

[25] Rama Danial Jhons Wenyi Zhao, Arvind. Discriminant analysis of principal component for face recognition.

[26] Wikipedia. Tensor.