International Journal of Geographical Information System Applications and Remote Sensing

"المجلة الدولية لتطبيقات نظام المعلومات الجغرافي والاستشعار عن بعد"

Vol. 2, No. 1, May 2011
ISSN **2077-771X***; Copyright © ICSRS Publication, 2011*
www.i-csrs.org

Hand Image Verification Method Based on PCA Eigenvectors

S.M. Ali and H.A. Hameed

Remote Sensing Research Unit, College of Science, University of Baghdad, Iraq, Baghdad, Al-Jaderyia
e mail: Sensing.remote@gmail.com

Abstract

In this research, the Principal Component Analysis (PCA) based method (namely Eigenhand), is used to verify persons from their hand's image. Our approach treats the hand recognition/verification problem as an essentially 2D-problem rather than requiring recovery of 3D geometry, taking advantage of the fact that hand's images can be described by a small set of 2D characteristics features. Here, the features are referred as "Eigenhands" because they represent the eigenvectors of the set of the trained and tested hands. The verification operation between the trained hand's images (i.e. preserved in the Database) and the input "unknown" hand image is performed by utilizing the Minimum-Mean-Distance "MMD" criterion. Several amounts of different noises (i.e. Gaussian, Uniform, Salt-and-pepper) have been added to the test hand to measure the reliability of our presented verification system in the presence of the noise. Invariant tests against image rotation, resizing, shifting have also been carried.

Keywords*: Hand verification, Hand recognition, Hand identification.*

1. Objective of the paper:

The goal of this paper is to design a computer system capable for recognizing, identifying, and verifying humans from their hand's images, using the

Eigenvectors of the Karhunen-Loève –Transformation "KLT" or the Principal Component Analysis "PCA". The Eigenvectors are computed from the Covariance matrix of the set of trained and test hand's images. The system is designed as to be applicable on certain degrees of image corrupted phenomena; i.e. ***scaling, rotation, shifting and different types of noises*** (Gaussian, salt-and-pepper and uniform noise), using suitable image preprocessing techniques.

2. Introduction

Fingerprints recognition, face recognition, and iris recognitions have been active research areas over the past 40 years. They have been studied by scientists from different areas of psychophysical and computer sciences. The psychologists and neurosciences mainly deal with the human perception part of the topic, while engineers deal with machine recognition of studied features, [1]. In fact, the hands recognition application, as well as the faces, irises, and fingerprints have wide applications in the fields of security, biometric, access control, and surveillance systems. Biometric applications include; face recognition, fingerprint identification, iris identification, voice recognition, signature recognition, retina identification, and hand geometry identification [2].

Most of the attempts to identify the hands have been restricted environmentally; e.g. hand's skin color is surprisingly uniform and, therefore, utilizing color-based methods for detecting hand's type could be possible, for instance see [3-5]. In fact, little works have been done on hand's recognition on monochromatic images, they mostly based on their appearance and their textures. As an example, Wu and Huang [6], have investigated the suitability of a number of classification methods to prove the view independence of hand posture recognition; the objective was to perform hands classification without considering their skin color. However, the best pattern recognizers in most instances are humans, yet we do not understand how humans recognize patterns, [7].

One of the simplest and earliest approaches to pattern recognition are those based on template matching (used to determine the similarity between two entities of the same type). The similarity measure may be performed by determining the cross-correlation-coefficient between the matched templates, [8].

Other widely used approaches were those utilized statistical features. The goal of these recognition methods is to choose features that allow pattern vectors belonging to different categories to occupy compact and disjoint regions in a d-dimensional feature space, [9, 10]. The effectiveness of these representations is determined by how well patterns from different classes can be separated. Statistical features may include probabilistic measures (e.g. mean, variance/standard deviation, and entropy values), they may be extracted from the 2D-histograms (i.e. the Co-Occurrence matrices), [11]. Moreover, the statistical methods may include measuring the correlation, the principal component analysis (PCA), the linear discrimination analysis (LDA), etc, [12, 13].

Transform based approaches have been proposed as alternative recognition methods utilizing frequency coefficient features; e.g. Discrete Cosine Transform DCT, Discrete Fourier Transform DFT, Discrete Wavelet Transform DWT are examples of these transformation approaches [14-16]. Neural network approaches have also been used in pattern recognition generally in a geometrical local feature based manner, [17]. In this recognition system, the obtained fractal codes are fed as inputs to a back-propagation neural network for identifying an individual.

Practically, structural or texture based approaches have proved its superiority in yielding best classification and recognition results in varying image environments. This recognition approach is, in fact, intuitively appealing because, in addition to recognition ability, it provides a description of how the given pattern is constructed from the primitives, [18, 19]. The method presents in this paper can be consider as to be hybrid recognition technique which based on both synthetic and structural features.

3. Preprocessing Operations:

The design of a pattern recognition system essentially involves the following three aspects: data acquisition and preprocessing, data representation, and decision making. The preprocessing step includes segmenting/isolating the object of interest from the background using grey tone thresholding/edge detection methods, image noise reduction using spatial/frequency domain filters, image contrast enhancement methods using histogram equalization/modification methods, etc.

3.1 Image segmentation by thresholding and/or edge detection processes:

Image segmentation is an essential preliminary step in most automatic pictorial pattern recognition and scene analysis problems. It is a fundamental tool in image processing and computer vision, particularly in the areas of feature detection and feature extraction, which aim to recognize and/or identify certain patterns.

Because of its intuitive properties and simplicity of implementation, image thresholding play a central position in applications of image segmentation. Thresholding may be viewed as an operation that involves tests against a function T of the form [20];

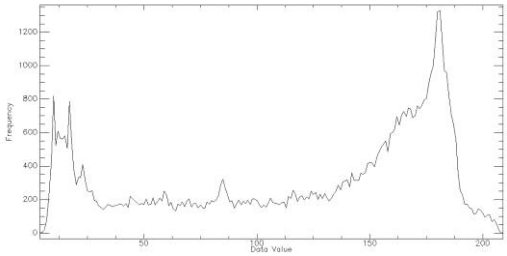
$$T = T[x, y, \mu(x, y), f(x, y)] \tag{1}$$

where $f(x, y)$ is the gray level of point (x, y) and $\mu(x, y)$ denotes some local property of this point—for example, the average gray level of a neighborhood centered on (x, y). A thresholded image $g(x, y)$ is defined as;

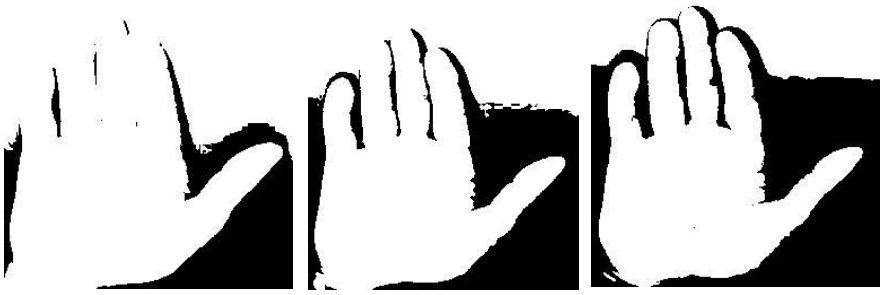
$$g(x, y) = \begin{cases} 1 & if \quad f(x, y) > T \\ 0 & if \quad f(x, y) \le T. \end{cases} \tag{2}$$

As an example, see Fig.(1), the original hand image, its histogram, and its thresholded versions (T=50, 100, and 128, respectively).

a) Original image



b) Image histogram



c) T=50 d) T=100 e) T=128

Figure (1): a) original hand image, b) its histogram, c-e) thresholded versions with different threshold values (T = 50, 100, and 128).

Image segmentation algorithms generally are based on one of two basic properties of intensity values: i.e. ***discontinuity*** and ***similarity***. In the first category, the approach is to partition an image based on abrupt changes in intensity, such as ***edges*** in an image, while the approaches in the second category are based on partitioning an image into regions that are similar according to a set of predefined criteria. Thresholding (discussed above), region growing, and region splitting and merging are examples of methods in this category, [20]. However, the approaches used to detect discontinuities in image gray values are, usually, referred as edge detectors. Several techniques for detecting edges have been developed and used to delineating regions within an image. The simplest types of these detectors are based on implementing differential/gradient "∇" operator and

assigned all differentiated points having values above certain threshold to as edges. For an image function *f(x,y)*, the differentiated image *g(x, y)*is given by;

$$g(x,y) = |\nabla f(x,y)| = \sqrt{\left(\frac{df(x,y)}{dx}\right)^2 + \left(\frac{df(x,y)}{dy}\right)^2} \quad (3)$$

Examples of the most commonly used gradient masks are shown in Fig.(2). The problem with the differential/gradient edge detection techniques lies in their dependent on threshold value; i.e. utilizing large threshold producing disconnected boundaries, while using small threshold value creates thick boundaries (i.e. undesired edge points are also detected), as illustrated in Fig.(3).

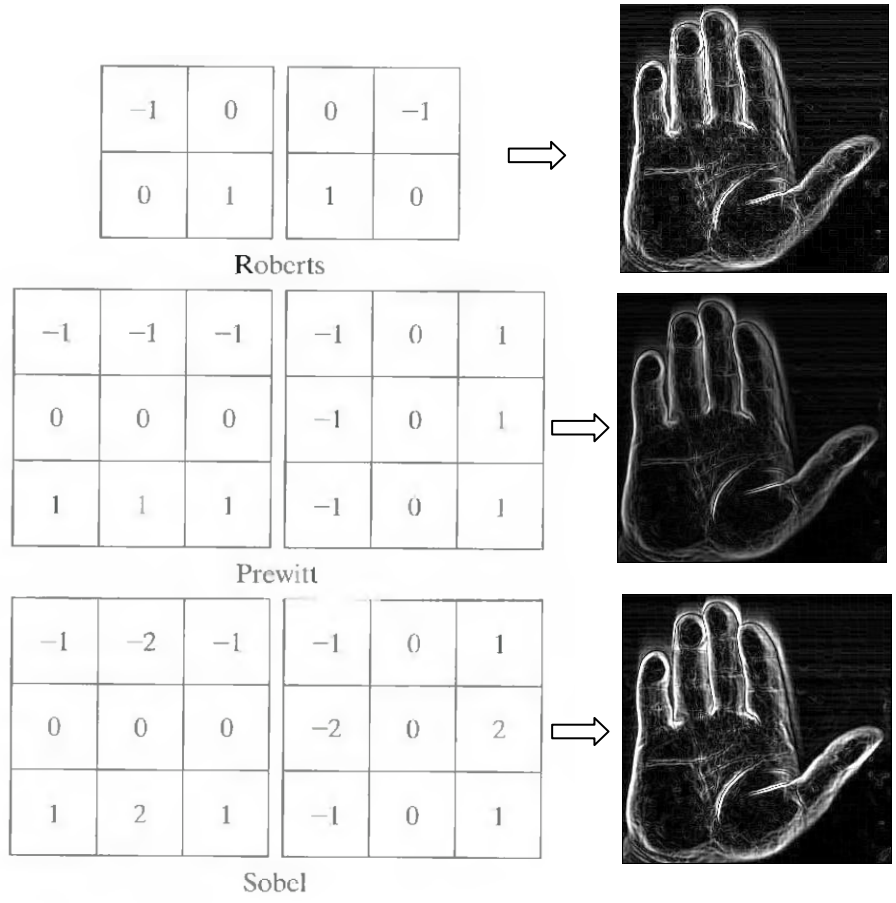
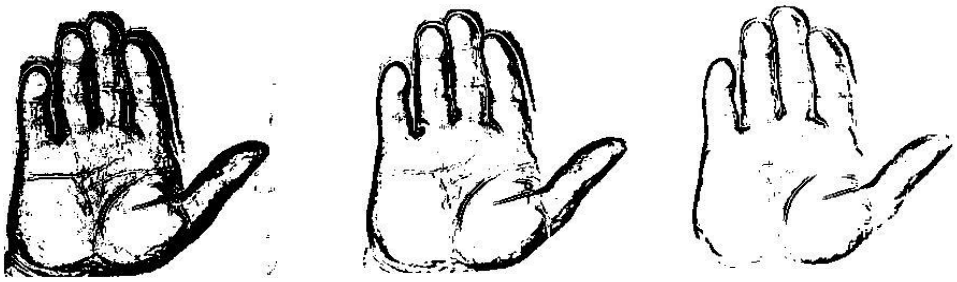


Figure (2): Digital representation of the most common used gradient operators (left), their performances on the hand image (right).



T=50 T=100 T=150

Figure (3): Thresholded versions of Sobel gradient image, using different threshold values.

To overcome the threshold dependency problems associated with the gradient edge detectors, several edge techniques have been introduced based on convoluting the image with a band-pass filter (e.g. Laplacian of Gaussian Operator), detecting the zero-crossing points as edge points. Examples of these edge detectors are Marr-Hildreth [21], Geuen [22], and Canny techniques [23]. The edge detection processes by these techniques is illustrated in Fig. (4).

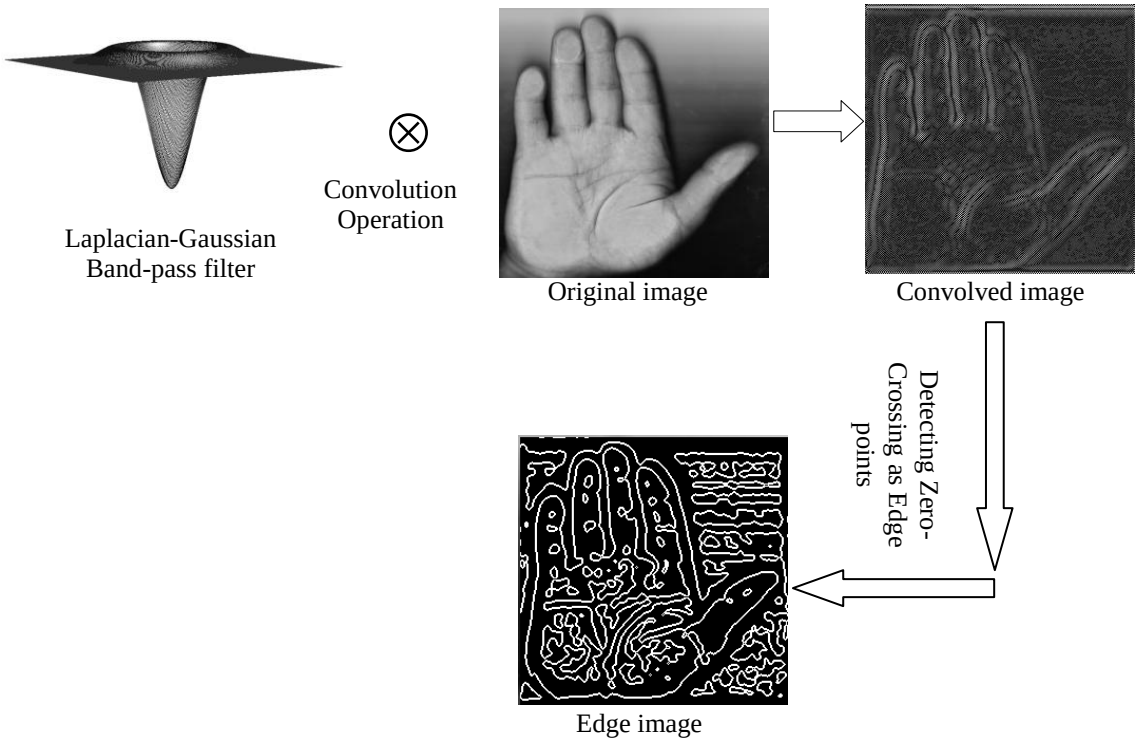


Figure (4): Demonstrating the main edge detection processes used in the band-pass (zero-crossing) techniques.

3.2 Image Enhancement Processes:

The idea behind enhancement techniques is to highlight certain features of interest in an image, or simply may be used to reduce the effects of certain undesired signal or noise within the image. The principal sources of degradation (e.g. noise, bad illumination, camera motion, etc) in digital images arise during image acquisition (digitization) and/or transmission. For instance, the major factors affecting the amount of noise during acquiring images with a CCD camera are the light levels and sensor temperature. The corruption in images during transmission is due to interference in the channel used for transmission. For example, an image transmitted using a wireless network might be corrupted as a result of lightning or other atmospheric disturbance [20].

For example, the hand image used in figures above has been badly illuminated during image acquisition, therefore the edge image in Fig.(4) showed undesired edge lines at the image's background. To overcome this problem, image thresholding may be used to assign as edges only those points having zero-crossing values (in the band-pass image) larger than certain desired threshold value "T_1". Example of edge images, obtained by implementing Canny's method with filter's standard deviation σ=2 and different zero-crossing threshold values, are shown in Fig.(5). During acquisition, the image may be blurred by planar motion caused by camera movement. If *D* is the duration of the camera exposure, and $x_0(t)$ and $y_0(t)$ be the time-varying components of the motion in the *x* and *y* directions. The output blurred image by uniform linear motion, thus, can be represented by [24];

$$g(x,y)=\int_0^D f[x-x_0(t),y-y_0(t)]dt \tag{4}$$

Where $f(x, y)$ is the unaffected image

The motion components can be estimated by the zeros of the Fourier transform of the blurred image function; i.e.

$$G(u,v)=F(u,v)H(u,v)=F(u,v)\int_0^D \exp^{-j2\pi[ux_0(t)+vy_0(t)]}\,dt \tag{5}$$

Where; $x_0(t)=Xt/T$ and $y_0(t)=Yt/T$ represent the rates of motion in the *x* and *y* directions, *X* and *Y* represent the total distances of image displacement in the *x* and *y* directions.

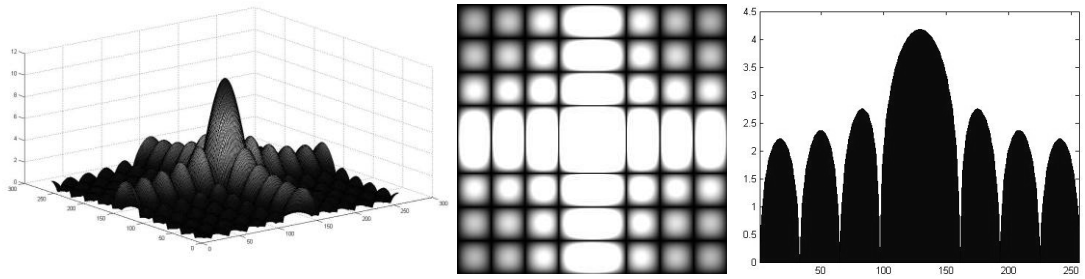
The ***spectrum*** of the blurring function in eq.(5), can be proven to be;

$$|H(u,v)|=DXY\left|\frac{\sin(\pi uX)}{\pi uX}\right|\left|\frac{\sin(\pi uY)}{\pi uY}\right| \tag{6}$$

Figure (6) shows three-dimensional (3D), two-dimensional (2D), and one-dimensional (1D) views of the Fourier spectrum for the blurring function $|H(u, v)|$, illustrating the zeroes that are used to estimate the motion components.

T_1=0.0 T_1=0.1 T_1=0.2

Figure (5): Edge images obtained by Canny's method (σ=2) with different illustrated zero-crossing threshold values.



a) 3D-plot of the |H(u, v)| b) 2D-image of |H(u, v)| c) 1D-plot of the |H(u, v)| central line

Figure (6): multidimensional views for the Fourier spectrum |H(u, v)| of the motion cause blurring function.

The *X* and *Y* components of the motion can now be estimated from the zeroes position on the horizontal and vertical frequency coordinates of the Fourier spectrum of |H(u, v)|; i.e. $X = round(\frac{v_n}{4n})$ and $Y = round(\frac{u_n}{4n})$, where u_n & v_n represent, respectively, the n[th] zero's positions on the *u* & *v* axes.

As the motion components X & Y are estimated, the blurred image by the motion can be de-blurred, using any de-blurring filter (as an example, the parametric Wiener filter), for instance see [20]. Generally, degraded image is presented in the spatial domain by;

$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y) \quad (7)$$

The symbol "*" indicates convolution.

In the frequency domain, eq.(7) is represented by;

$$G(u,v) = H(u,v)\,F(u,v) + N(u,v) \quad (8)$$

As mentioned above, H(u, v) represents the blurring function caused by certain degradation phenomenon, while $\eta(x, y)$ or, alternatively $N(u,v)$ in the frequency domain, represents the principal sources of noise that may be arisen during image acquisition (digitization) and/or transmission operations. Too many types of different noises are existed; the following represent the most common of them:

- ***The Gaussian Noise***: it is also called ***normal*** noise model, the Probability Density Function "PDF" of the Gaussian noise is-given by:

$$\rho(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \tag{9}$$

Where *z* represents the image gray level, μ is the mean of the average value of *z*, and σ is its standard deviation.

- ***The Rayleigh Noise***: the PDF of this noise is given by

$$\rho(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & \text{for } z \geq a \\ 0 & \text{for } z < a \end{cases} \tag{10}$$

The mean and variance of the Rayleigh noise is given by;

$$\mu = a + \sqrt{\frac{\pi b}{4}} \quad \text{, and} \quad \sigma^2 = \frac{b(4-\pi)}{4}$$

- ***The Uniform Noise***: it has the following PDF

$$\rho(z) = \begin{cases} \frac{1}{b-a} & \text{if } a \leq z \leq b \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

It has the mean and variance values

$$\mu = \frac{a+b}{2} \quad \text{, and} \quad \sigma^2 = \frac{(b-a)^2}{12}$$

- ***The Impulse (Salt and Pepper) Noise***: The PDF of this (bipolar) impulse noise is given by

$$\rho(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

If $b>a$, gray-level b will appear as a light dot in the image (i.e. Salt noise mode), conversely, level a will appear as dark dot (i.e. Pepper noise mode) if either P_a or P_b is zero (in such case the noise is called ***unipolar***). The PDFs of the above mentioned noises are illustrated in Fig.(7).

For the past three or four decades too many digital filtration methods (in both the spatial and frequency domains) have been developed to reduce the effects of the noise within the images. The most common used spatial domain's filters are the mean and order-statistics filters, while the well known frequency's filters are the low-pass and high-pass filters, for the details see Gonzalez and Woods [20].

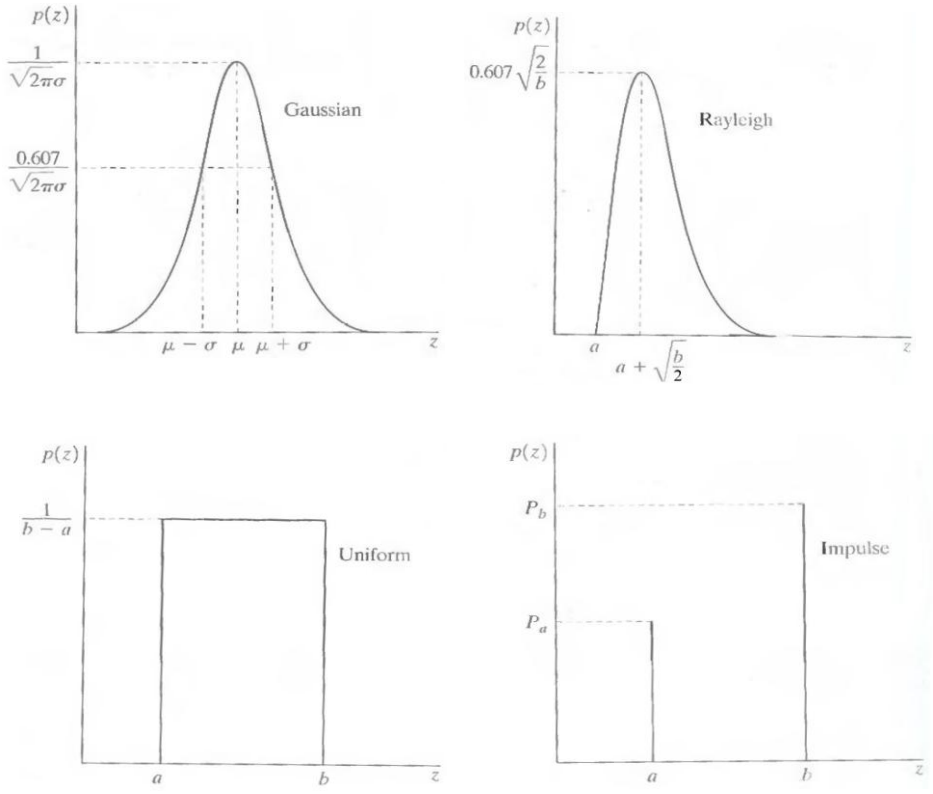


Figure (7): PDFs of several most common practical noise models.

4. Pattern Classification Basics:

Pattern recognition is the science that concerns the description or classification of measurements. A typical pattern recognition system takes the input data, extracts the features, trains the classifier and evaluates the test pattern. There are three main pattern recognition approaches; i.e. ***statistical***, ***syntactic***, and

neural pattern recognition approaches [7]. In this research, statistical pattern approach will be used.

The statistical pattern recognition is an approach of the pattern classification problem, which concentrates on developing decision or classification strategies that form classifiers; i.e. each pattern is represented in terms of *d* features or measurements and, thus, can be viewed as a point in a *d*-dimensional space [1]. The objective of a statistical pattern recognition system is to choose features that allow the patterns belonging to different classes to occupy different regions in a *d*-dimensional feature space. The representation space or feature space is assumed to be effective, if the patterns from different classes are well separated and if the feature space greatly represents the general properties of the input data. Decision boundaries separate patterns belonging to different classes and they are determined by the specified probability distributions of the patterns belonging to each class. For the case where the training samples used to design a classifier are labeled by their category membership, the procedure is accepted as ***supervised***. The opposite case is called ***unsupervised*** system where the training data is unlabeled.

4.1. Feature Dimension Reduction:

Feature selection involves the derivation of certain features from the input data in order to reduce the amount of data used for recognition and provide discrimination power. Due to the measurement cost and recognition/ verification accuracy, the number of features should be kept as small as possible. Less number of selected features makes the system work faster and uses less memory. However, feature extraction methods try to reduce the feature dimensions used in the recognition step. One of the most common used methods in pattern recognition to reduce the feature dimensions is the PCA, [7]. The advantage of PCA comes from; it reduces the feature space dimension by considering the variance of the input data. In order to obtain the best variance in the data, the data is projected to a subspace (of the image space) which is built by the ***eigenvectors*** from the data. In that sense, the eigenvalue corresponding to an eigenvector represents the amount of variance that eigenvector handles. The mathematical formulation of PCA is discussed below.

5. Eigenhand Recognition Based on KLT Method:

The best known linear feature extractor is the ***PCA*** or Karhunen-Loève (***KL***) expansion that computes the largest eigenvectors of the covariance matrix of the patterns [7]. As an example, let we have ***m*** hand-images, each of size $N \times N$ pixels. Each sample image $f_k(x, y)$ can be expressed in the form of an N^2-dimensional vector $\boldsymbol{X}_k$, which constructed by sticking image rows after other (it can also be formed by using columns instead of rows), as follows:

$$\boldsymbol{X}_k = \begin{bmatrix} f_{k(1,1)} \\ f_{k(1,2)} \\ . \\ f_{k(1,N)} \\ f_{k(2,1)} \\ . \\ f_{k(2,N)} \\ . \\ . \\ . \\ f_{k(N,N)} \end{bmatrix} \tag{13}$$

Where $f_{k(i,j)}$ denotes the (i, j)th element of the kth image.

Let $\boldsymbol{M}$ represents the matrix involves $\boldsymbol{m}$ vectors of the hand-images; i.e.

$$\boldsymbol{M} = \begin{bmatrix} f_{1(1,1)} & f_{2(1,1)} & . & . & f_{m-1(1,1)} & f_{m(1,1)} \\ f_{1(1,2)} & f_{2(1,2)} & . & . & f_{m-1(1,2)} & f_{m(1,2)} \\ . & . & . & . & . & . \\ f_{1(1,N)} & f_{2(1,N)} & . & . & f_{m-1(1,N)} & f_{m(1,N)} \\ f_{1(2,1)} & f_{2(2,1)} & . & . & f_{m-1(2,1)} & f_{m(2,1)} \\ . & . & . & . & . & . \\ f_{1(2,N)} & f_{2(2,N)} & . & . & f_{m-1(2,N)} & f_{m(2,N)} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ f_{1(N-1,N)} & f_{2(N-1,N)} & . & . & f_{m-1(N-1,N)} & f_{m(N-1,N)} \\ f_{1(N,N)} & f_{2(N,N)} & . & . & f_{m-1(N,N)} & f_{m(N,N)} \end{bmatrix} \tag{14}$$

The covariance matrix of $\boldsymbol{M}$ can now be defined by;

$$\boldsymbol{C_x = (M - \overline{X})^T (M - \overline{X})} \tag{15}$$

Where: '^{T}' indicates transposition, and;

$$\boldsymbol{\overline{X} = \frac{1}{m} \sum_{i=1}^{m} X_i}$$

Since the matrix $\boldsymbol{(M - \overline{X})}$ has the dimensionality N^2×m, the covariance matrix will have a dimensionality of m×m. Let e_i and λ_i, i=1,2,…, m be the eigenvectors and eigenvalues of C_x. For convenience, it is assumed that the eigenvalues have been arranged in decreasing order, i.e. $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. The matrix

whose rows are the eigenvectors of C_x represents the *KL*-transformation matrix, given by;

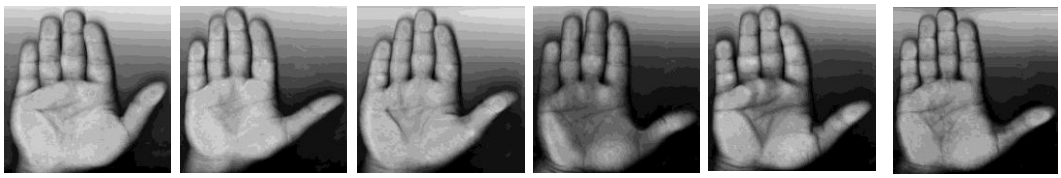
$$A = \begin{bmatrix} e_{11} & e_{12} & . & . & . & e_{1m} \\ e_{21} & e_{22} & . & . & . & e_{2m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ e_{m1} & e_{m2} & . & . & . & e_{mm} \end{bmatrix} \tag{16}$$

The *KL*-transformation is simply performed by multiplying the centralized image vectors $(X_i - \overline{X})$ by *A* to obtain a new image vector; i.e.,

$$Y_i = A(X_i - \overline{X}) \tag{17}$$

6. The Experimental Results:

For the purpose of simplicity, we shall start with as fewer numbers of samples as possible. Let us adopt first only 6-samples of hand images shown in Fig.(8).



(1) (2) (3) (4) (5) (6)

Figure (8): Training set of six hand images, each of 256×256 pixels size.

The average hand's image is shown in Fig.(9).

Figure (9): The average hand of the training set.

Figure (10) represent the mean reduced (each referred as ***Eigenhand)*** of the training set of the hand's images.

(1) (2) (3) (4) (5) (6)

Figure (10): The mean reduced ***Eigenhand***s of the training set of images.

The computed eigenvalues and eigenvectors are given in Table-1. Note, for convenience the eigenvalues and eigenvectors elements are approximated to nearest four decimal values.

Table-1: The Eigenvalues and Eigenvectors of the trained images

Eigenvalues λ's	V_1	V_2	V_3	V_4	V_5	V_6
1.7595e+008	0.4083	0.4083	0.4083	0.4083	0.4083	0.4083
8.0402e+007	-0.2388	-0.5440	0.3134	0.7186	-0.0978	-0.1513
3.6076e+007	0.0344	-0.2979	0.1715	-0.1924	-0.4915	0.7760
3.1982e+007	0.3392	0.0552	-0.7566	0.4786	-0.2501	0.1336
2.5753e+007	-0.5926	0.6515	0.0681	0.2232	-0.4082	0.0580
-1.4901e-008	-0.5558	-0.1457	-0.3587	0.0334	0.5942	0.4326

Let us now see what would happen if one of the above trained images is used as a test image; i.e. the set of the trained image will be increased by inserting a test image. Table-2, illustrates the computed Eigen values and vectors of the new trained plus test set of images.

Table-2: The Eigenvalues and Eigenvectors of the trained and test images

Eigenvalues λ's	V_1	V_2	V_3	V_4	V_5	V_6	V_7
1.9823e+008	-0.3777	-0.3777	-0.4055	-0.3777	-0.3777	-0.3777	-0.3498
8.0827e+007	0.0149	0.0149	-0.6917	0.0149	0.0149	0.0149	0.7214
5.2917e+007	0.1815	0.5112	-0.0830	-0.8063	0.1022	0.1774	-0.0830
3.2134e+007	-0.0801	0.3088	-0.0272	0.0785	0.5289	-0.7816	-0.0272
2.6448e+007	-0.4584	-0.3396	0.4849	-0.3911	0.2264	-0.0071	0.4849
3.1148e-008	-0.6487	0.6159	0.0771	0.1946	-0.3818	0.0657	0.0771
-4.8470e-007	0.4323	0.0763	0.3293	-0.1002	-0.6082	-0.4587	0.3293

As it is obvious, the similarity between the test image (represented by the eigenvector V_7) and the trained set is presented by the 3rd eigenvector V_3. The

similarity test can be performed by utilizing the Minimum-Mean-Distance "*MMD*" criterion given by [9]:

$$MMD = Min\{\sum_{i=1}^{m-1}\sum_{k=1}^{m}|V_{m,k} - V_{i,k}|\} \quad (18)$$

Where: "*i*" represents the vector indexing number, and "*k*" is the vector's elements number. As an example, Table-3 represents the counted MMD values for the eigenvectors given in Table-2, and by considering different number of vector's elements.

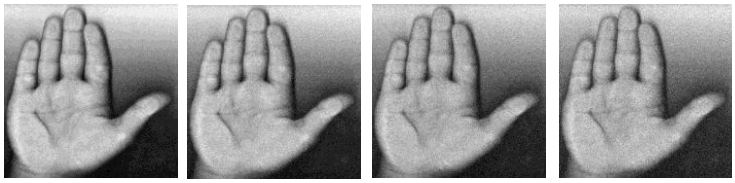
Table-3: The counted MMD between the test and trained images.

No. of adopted elements (in descended order)	$\|V_7 - V_1\|$	$\|V_7 - V_2\|$	$\|V_7 - V_3\|$	$\|V_7 - V_4\|$	$\|V_7 - V_5\|$	$\|V_7 - V_{61}\|$
6	2.8615	3.3185	1.5441	3.024	3.1683	3.0782
5	2.7932	3.2503	1.4076	2.9558	3.1001	3.01
4	2.0894	2.5464	0.0	2.252	2.3963	2.3062

Let us now examine the effects of the noise on the similarity tests between the eigenvectors of the image to be verified and the trained set of images. Figure (11) demonstrates the hand image (shown in Fig.(8-3) with varied amounts of different noise. The noise effects test is illustrated in Table-4, by comparing the eigenvectors of the noisy images with that of the test image listed as $\boldsymbol{V_7}$ in Table-2. The Minimum-Mean-Distance "MMD" between the test and trained set of images, for varying amount of different noise, are listed in Table-5. The validity of the verifying test of the introduced method is illustrated in Fig.(12) which represent the plots of the MMD values against the noise's amounts (presented by the noise's standard deviations or by the probability density function).

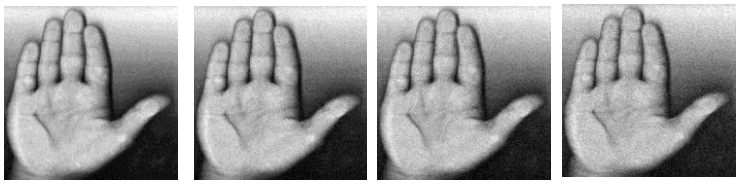
Table-4: The Eigenvectors of the noisy images

Eigenvector		**Eigenvector elements**						
Non noisy image		-0.3498	0.7214	-0.0830	-0.0272	0.4849	0.0771	0.3293
Image with Gaussian noise	σ=7	-0.3780	-0.6470	-0.1358	-0.0273	0.4297	0.0885	0.4763
	σ=10	0.3780	0.6579	0.1362	-0.0282	-0.4401	0.0656	0.4550
	σ=12	0.3780	0.6788	0.1196	-0.0293	-0.4627	0.0436	0.4064
	σ=15	0.3780	-0.6360	0.1497	-0.0299	0.4793	0.0342	0.4454
Image With Uniform noise	σ=7	0.3780	0.6207	0.1427	-0.0282	-0.4289	0.1034	0.5060
	σ=10	0.3780	0.6238	0.1418	-0.0286	-0.4454	0.0969	0.4893
	σ=12	0.3780	0.6164	0.1458	-0.0301	-0.4571	0.0941	0.4871
	σ=15	0.3780	-0.6056	0.1551	-0.0310	0.4753	0.0753	0.4835
Image With Impulse Salt-Pepper noise	ρ_a, ρ_b =0.03	0.3780	-0.3032	0.2026	0.0435	0.3906	0.6054	0.4507
	ρ_a, ρ_b =0.04	0.3780	0.1797	0.2145	0.0571	0.0978	0.7147	0.5051
	ρ_a, ρ_b =0.05	0.3780	-0.1114	0.1891	0.0732	0.0407	0.6954	0.5643
	ρ_a, ρ_b =0.08	0.3780	-0.0358	0.0835	-0.1072	0.0031	0.5449	0.7352



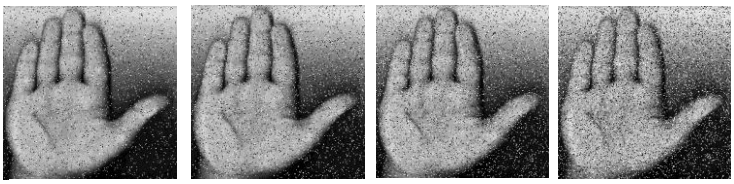
σ=7 σ=10 σ=12 σ=15

c) Additive Gaussian noise with mean=0, and indicated standard deviation values.



σ=7 σ=10 σ=12 σ=15

a) Additive Uniform noise with mean=0, and indicated standard deviation values.



ρ_a=0.03 ρ_a=0.04 ρ_a=0.05 ρ_a=0.08

b) Additive Impulse Salt-Pepper noise with indicated probability values.

Figure (11): Corrupted images with different amount of different noise.

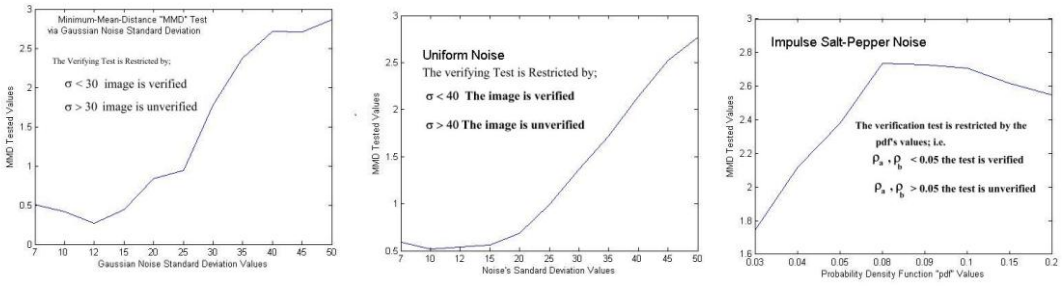


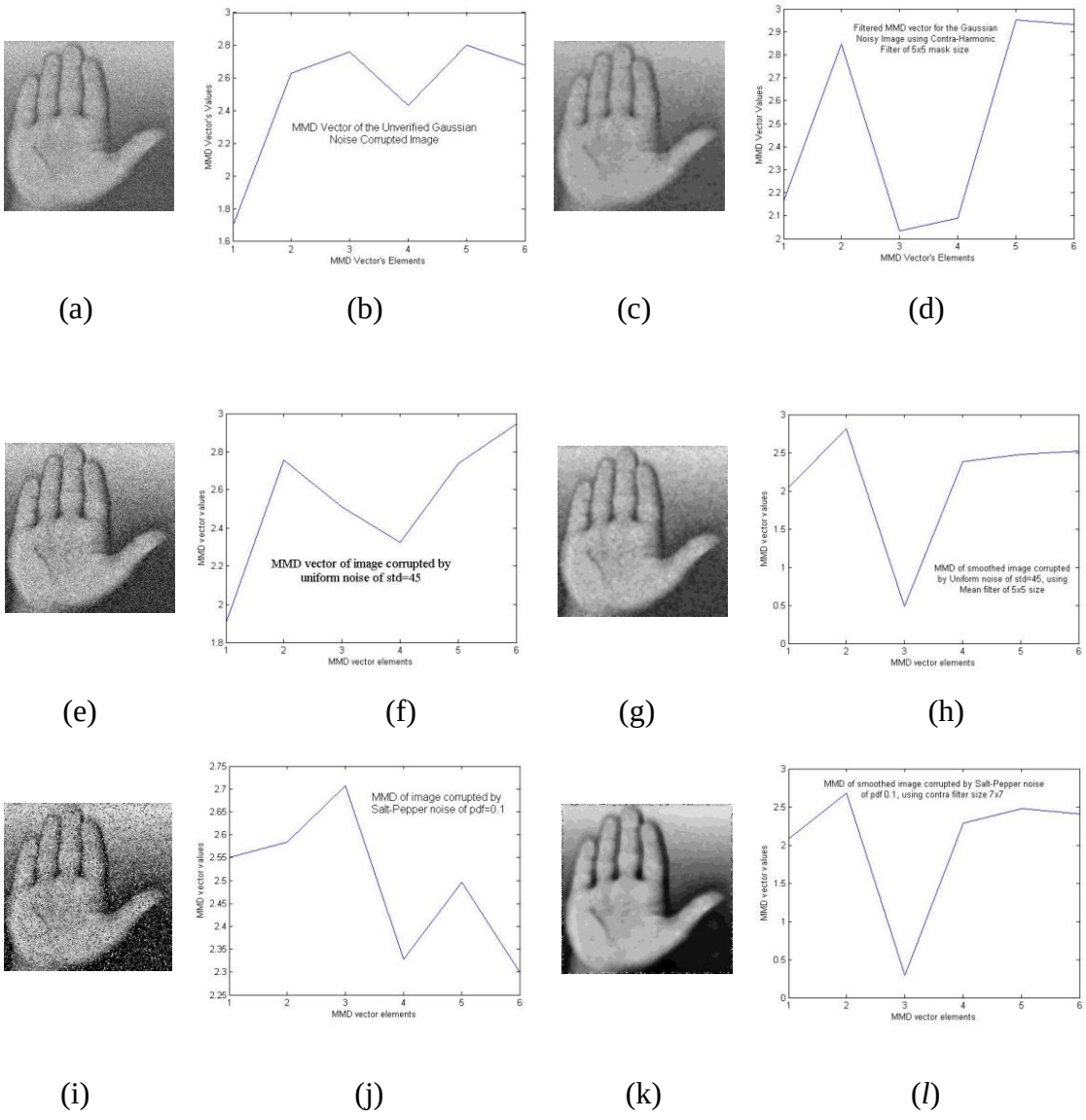
Figure (12): Plots of the MMD via the noise's amount of noise, indicating the verification condition as a function of the amount of each types of noise.

Table-5: The counted MMD between the test and trained images.

No of adopted elements= 5	$V_7 - V_1$	$\lvert V_7 - V_2 \rvert$	$\lvert V_7 - V_3 \rvert$	$\lvert V_7 - V_4 \rvert$	$\lvert V_7 - V_5 \rvert$	$\lvert V_7 - V_6 \rvert$	*Test result*
Gaussian σ=7	٢,١٣٨٩	٢,٧٣٩٤	٠,٥٠٢٥	٢,٢٣٨٨	٢,٤١٠٨	٢,٤١٦٣	Verified
Gaussian σ=10	2.0892	2.7353	0.4164	2.2751	2.4219	2.4065	Verified
Gaussian σ=12	1.9897	2.6870	0.2634	2.3159	2.4220	2.3897	Verified
Gaussian σ=15	2.0881	2.7894	0.4457	2.3450	2.4741	2.4668	Verified
Uniform σ=7	2.2075	2.7760	0.5900	2.2183	2.4145	2.4486	Verified
Uniform σ=10	2.1873	2.7733	0.51807	2.2376	2.4339	2.4481	Verified
Uniform σ=12	2.1945	2.7880	0.5379	2.2469	2.4523	2.4572	Verified
Uniform σ=15	2.1948	2.8205	0.5596	2.2857	2.4818	2.4617	Verified
Salt-Pepper ρ_a, ρ_b =0.03	1.9436	2.8633	1.7427	2.6700	2.6239	2.9543	Verified
Salt-Pepper ρ_a, ρ_b =0.04	2.1308	2.8756	2.1179	2.4190	2.7961	2.8639	Verified
Salt-Pepper ρ_a, ρ_b =0.05	2.2112	2.7382	2.3794	2.3494	2.7066	2.8110	Unverified
Salt-Pepper ρ_a, ρ_b =0.08	2.4329	2.6256	2.7347	2.3876	2.5465	2.5506	Unverified

Note: The verification test is failed when the amount of noise is increased above $\sigma \geq 30$ for the Gaussian, $\sigma \geq 40$ for the Uniform, and when $\rho_a, \rho_b \geq 0.05$ for the Salt-Pepper noise, as illustrated in Fig. (12).

In fact unverified noisy images can be verified by subjecting them to certain suitable filtration method. Figure (13) illustrates selected versions of unverified samples of hand images and their filtrated versions. As is obvious, the first column represents the corrupted version of images by, respectively, Gaussian (σ = 30), Uniform (σ = 45), and Salt-Pepper (ρ_a, ρ_b =0.1), while the second column represents the unverified MMD vectors of the corrupted image. The third column represents the smoothed version of images using, respectively, Gaussian filter of size 5×5, Mean filter of size 5×5, and Contra-Harmonic filter of size 7×7, the verified MMD vectors of the smoothed images are shown in the forth column, for the details of the used filters see [20].



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (*l*)

Figure (13): demonstration of the smoothing filtration effects on the verification decision of the introduce eigenhand verification system.

$\theta=1^o$ $\theta=2^o$ $\theta=3^o$ $\theta=4^o$ $\theta=5^o$

Figure (14): Rotated versions of hand image.

Table-6: The counted MMD vectors between trained and rotated images.							
Clockwise rotation angle "θ"	MMD vector's elements						Verification testing Result
1 °	2.1561	2.5781	0.2633	2.2481	2.4878	2.3167	Verified
2 °	2.2391	2.6116	0.5898	2.2334	2.5994	2.3672	Verified
3 °	2.3197	2.5972	1.0029	2.3688	2.7108	2.4336	Verified
4 °	2.3454	2.4562	1.4944	2.5749	2.7726	2.4826	Verified
5 °	2.1086	2.0571	2.7657	2.1752	2.6510	2.9915	Unverified

3% 6% 9% 10% 12%

Figure (15): Resized versions of hand image by the shown factor below each.

Table-7: The counted MMD vectors between trained and resized images.							
Resize Factor	**MMD vector's elements**						**Verification testing Result**
3%	**2.1818**	**2.5798**	**0.5955**	**2.169**	**2.4844**	**2.428**	**Verified**
6%	**2.3228**	**2.7803**	**0.7748**	**2.0816**	**2.7356**	**2.6102**	**Verified**
9%	**2.1847**	**2.7021**	**1.8778**	**2.0141**	**3.023**	**2.7280**	**Verified**
10%	**2.0401**	**2.6018**	**2.0138**	**2.0836**	**3.0051**	**2.6698**	**Verified**
12%	**2.0897**	**2.4012**	**2.2883**	**2.3187**	**2.7999**	**2.7392**	**Unverified**

1-pixel 2-pixels 3-pixels 4-pixels 5-pixels

Figure (16): Shifted versions of hand image presented by pixel size, shown below each.

Table-8: The counted MMD vectors between trained and resized images.							
Shift in pixels	MMD vector's elements						Verification testing Result
1	2.0243	2.5842	0.1453	2.3162	2.3833	2.3066	Verified
2	1.9440	2.6011	0.3210	2.3637	2.3615	2.3033	Verified
3	1.8378	2.5975	0.5412	2.4054	2.3266	2.2639	Verified
4	1.6948	2.5745	0.8225	2.4535	2.2773	2.1740	Verified
5	1.5671	2.5198	1.1814	2.5347	2.1978	2.0134	Unverified

Finally, it should be noted that; we have tested our verification system on a set larger samples 25 trained hand images, the results were the same as those achieved on the smaller examined set above. Figure (17) represents the verified MMD vector applied on a set of 25 trained hand images.

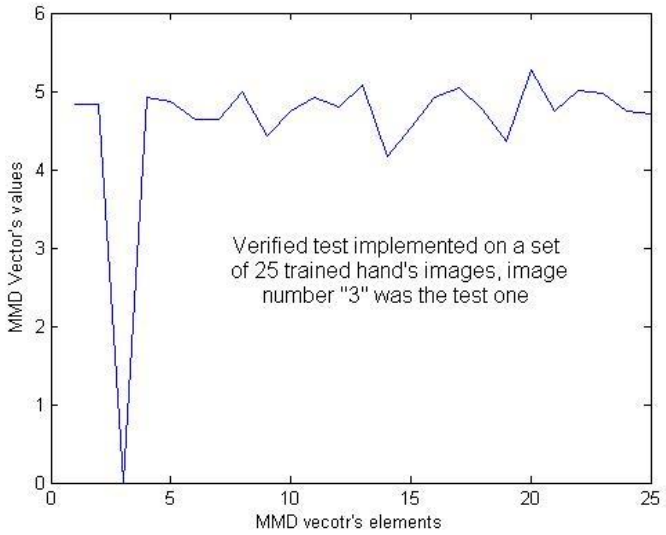


Figure (17): Verification test performed on a set of 25 images.

7. Conclusions:

A hand image verification system has been designed and introduced to identify the existence of an examined test image among a set of trained images. Several preprocessing and enhancement techniques have been discussed with the hope to be using them if they were required to improve the system's input images. The effects of noise, bad-illumination, resizing, rotation, and shifting have been studied and illustrated. The illustrated obtained results showed that the presented system is, to some extent, partially invariant toward the studied effects. In case if the test showed unverified result, some suitable manipulations have been suggested to improve the testing results.

References:

[1]. A.B. GUL, "Holistic face recognition by dimension reduction," M.Sc. thesis submitted to the graduate school of natural and applied sciences, Middle East Technical University, Istanbul-Turkey, (2003)

[2]. J. Huang, "Detection Strategies for face Recognition using Learning and Evolution," PhD Thesis, George Mason University, (1998)

[3]. D. Saxe, and R. Foulds, "Toward robust skin identification in video images," Proc. IEEE Intl. Conference on Automatic Face and Gesture Recognition, pp. 379-384, (1996).

[4]. M.J. Jones, and J.M. Rehg, "Statistical color models with application to skin detection," Int. Journal of Computer Vision, 46(1), pp. 81-96, (2002).

[5]. X. Zhu, J. Yang, and A. Waibel, "Segmenting hands of arbitrary color," Proc. IEEE Intl. Conference on Automatic face and Gesture Recognition, (2000)

[6]. Y. Wu, and T.S. Huang, "View-independent recognition of hand postures," Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2, pp.84-94, (2000)

[7]. A.K. Jain, R.P.W. Duin, and J. Mao, "Statistical Pattern Recognition: A Review," IEEE Trans. On Pattern Analysis and Machine Intel., Vol.22, No.1, pp. 4-37, (2000).

[8]. R. Bajcsy, and S. Kovacic, "Multi-resolution Elastic Matching," Computer Vision Graphics Image Processing, Vol. 46, pp.1-21, (1989)

[9]. L. Devroye, L. Gyorfi, and G. Lugosi, "A Probabilistic Theory of Pattern Recognition," Berlin: Springer-Verlag, (1996).

[10]. R.O. Duda, and P.E. Hart, "Pattern Classification and Scene Analysis," New York: John Wiley & Sons, (1973).

[11]. R.M. Haralick, "Decision Making in Context," IEEE, Trans. Pattern Analysis and Machine Intelligence, Vol.5, No.4, pp.417-418, (1983).

[12]. W. Zhao, A. Krishnaswamy, R. Chellappa, D.L. Sqets, and J. Weng, "Discriminant Analysis of Principal Component Components for Face Recognition," International Conference on Automatic Face and Gesture Recognition, (1998).

[13]. J. Daugman, "The importance of being random: Statistical principles of iris recognition," *Pattern Recognition*, **Vol.36**, No.2, pp. 279–291. (2003).

[14]. C. Podilchuk, and X. Zhang, "Face Recognition Using DCT based feature Vectors", IEEE, pp. 2144-2147, (1996).

[15]. H. Spies, and I. Ricketts, "Face Recognition in Fourier Space," Vision Interface '2000: pp. 38-44, Montreal, (2000).

[16]. B. Kepenekci, "Face Recognition Using Gabor Wavelet Transform", M.Sc. Thesis, METU, (2001).

[17]. P. Temdee, D. Khawparisuth, and K. Chamnongthai, "Face Recognition by Using Fractal Encoding and Back-propagation Neural Network," 5th Int. Symposium on Signal Processing and its Applications, ISSPA '99, pp. 159-161, Brisbane, Australia, August (1999).

[18]. Yanushkevich, S. N., Stoica, A. Srihari, S. N., Shmerko, V. P. and Gavrilova, M. L. Simulation of Biometric Information: The New Generation of Biometric Systems, In *Proc. Int. Workshop Modeling and Simulation in Biometric Technology*, Calgary, Canada, pp. 87–98, (2004)..

[19]. Yanushkevich, S. N., Stoica, A., Shmerko, V. P. and Popel, D. V. *Biometric Inverse Problems*, Taylor & Francis/CRC Press, Boca Raton, FL, (2005).

[20]. Gonzalez, R.C. and Woods, E.W., "Digital image processing," 2nd Edition, Prentice Hall, Inc., (2002).

[21]. Marr, D., and Hildreth, E., "Theory of Edge Detection." Proc. R. Soc. Land., Vol. B207, pp. 187-217, (1980).

[22]. Geuen, W.,"A fast edge detection algorithm matching visual contour perception in image sequence processing and dynamic scene analysis," NATO ASI series, by Huang, T.S., Vol.F2, (1983).

[23]. Canny, John, "A Computational Approach to Edge Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-8, No. 6, pp. 679-698, (1986).

[24]. Gonzalez, R.C. and Wintz, W., "Digital image processing," 2nd Edition, Addison-Wesley, Inc., (1987).