Arab J Sci Eng (2013) 38:3349–3366
DOI 10.1007/s13369-013-0654-6

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Gesture Feature Extraction for Static Gesture Recognition

Haitham Sabah Hasan · Sameem Binti Abdul Kareem

Received: 4 September 2011 / Accepted: 4 July 2013 / Published online: 3 September 2013

Abstract The goal of static hand gesture recognition is to classify the given hand gesture data represented by some features into some predefined finite number of gesture classes. This paper presents a novel technique for hand gesture recognition through human–computer interaction based on shape analysis. The main objective of this work is to explore the utility of two algorithms that are used for extracting the features to recognize the hand gestures. Artificial neural networks technique is used as a recognizer to assign these features to their respective hand gesture classes. The proposed system presents a recognition method to recognize a set of six specific static hand gestures, namely, Open, Close, Cut, Paste, Maximize, and Minimize. The hand gesture image is passed through three stages, preprocessing, feature extraction, and classification. In preprocessing stage, some operations are applied to extract the hand gesture from its background and prepare the hand gesture image for the feature extraction stage. In the first method, the hand contour is used as a feature which treats scaling and translation of problems (in some cases). The complex moment algorithm is, however, used to describe the hand gesture and treat the rotation problem in addition to the scaling and translation. The classification algorithm used is a multi-layer neural network classifier which uses back-propagation learning algorithm. The results show that the first method has a performance of 70.83 % recognition, while the second method, proposed in this article, has a better performance of 86.38 % recognition rate.

H. S. Hasan (✉) · S. B. A. Kareem
Department of Artificial Intelligence, Faculty of Computer Science and Information Technology, University of Malaya,
50603 Kuala Lumpur, Malaysia
e-mail: haitham@siswa.um.edu.my

Keywords Gesture recognition · Hand gestures · Artificial neural network · Human–computer interaction · Computer vision

الخلاصة

إن هدف تمييز إشارة اليد الثابتة هو لتصنيف بيانات إشارة اليد الثابتة المعطاة التي تمثلها بعض الميزات في بعض الأرقام المحدودة لطبقات الإشارة المحددة مسبقا. تعرض هذه الورقة العلمية تقنية جديدة لتمييز إشارة اليد من خلال التفاعل بين الإنسان والحاسوب بناء على تحليل الشكل، والهدف الرئيسي من هذا العمل هو استكشاف فائدة اثنتين من الخوارزميات التي تستخدم لاستخراج الميزات للتعرف على حركة اليد ، وقد استخدمت تقنية الشبكات العصبية الاصطناعية (ANNs) كمعرف لتعيين هذه الميزات لكل طبقات حركة اليد الخاصة بها ، ويقدم النظام المقترح طريقة للتعرف على مجموعة من ستة أنماط محددة ثابتة ومتحركة، مثل فتح، وإغلاق، وقص، ولصق، وتكبير وتصغير، حيث يتم تمرير صورة حركة اليد من خلال ثلاث مراحل، وهي تجهيزها واستخراج الميزة والتصنيف ، ويتم - في مرحلة التجهيز- تطبيق بعض العمليات لاستخراج حركة اليد من خلفيتها وإعداد صورة حركة اليد من أجل مرحلة استخراج الميزة ، يتم - في الأسلوب الأول - استخدام محيط شكل اليد كميزة تعالج مشاكل التوسع والترجمة (في بعض الحالات) ، ومع ذلك استخدمت خوارزمية عزم القوة المعقدة لوصف حركة اليد وعلاج مشكلة الدوران ، إضافة إلى التوسع والترجمة. إن خوارزمية التصنيف المستخدمة هي مصنف الشبكة العصبية متعدد الطبقة الذي يستخدم خوارزمية تعلم الانتشار الخلفي ، وبينت النتائج أن الأسلوب الأول لديه أداء تمييز بمقدار 70.83٪، في حين أن الأسلوب الثاني المقترح في هذه المقالة لديه معدل أداء تمييز أفضل بمقدار 86.38٪.

1 Introduction

Since the first introduction of computer into the modern era, it has penetrated into all corners of our personal and social lives as a key element revolutionizing our way of living. Surfing the web, typing a letter, playing a video game or storing, and retrieving personal or official data are just a few examples of the use of computers or computer-based devices.

Due to increase in mass production and constant decrease in price of personal computers, they will even influence our everyday life more in near future. Nevertheless, in order to efficiently utilize the new phenomenon, myriad number of studies has been carried out on computer applications and their requirement of more and more interactions. For this reason, human–computer interaction (HCI), among others, has been considered to be a lively field of research in the last few years [1].

Since the introduction of the most common input devices, they have not changed considerably, probably due to the fact that the existing devices are still sufficiently applicable and effective. However, it is also well known that computer have been tightly integrated into everyday life with the constant introduction of new applications and hardware in the recent years [2]. Nowadays, majority of the HCI is based on mechanical devices such as keyboard, mouse, joystick, or game-pad, but a growing interest in a class of methods based on computational vision has been emerged due to ability to recognize human gestures in a natural way [3]. Recently, use of human movements, especially hand gesture, has become an important part of human–computer intelligent interaction (HCII), which serves as a motivating force for research in modeling, analyzing, and recognizing the hand gestures. Many techniques developed in HCII can also be extended to other areas such as surveillance, robot control, and teleconferencing [4]. The significance of the problem can be easily illustrated by the use of natural gestures applied together with verbal and non-verbal communications [5].

1.1 Background Studies

Static gesture recognition is a pattern recognition interface. The first step, before using any standard pattern recognition technique, is feature extraction. In this recognition, features correspond to the most discriminated information contained in the lighting conditions. A fair amount of research has been done on different aspects of this approach. Some of the approaches and concepts in connection with gesture recognition are introduced and evaluated in this paper. In line with this trend, Freeman and Roth [6] have jointly introduced a method to recognize hand gestures, based on a pattern recognition technique developed by McConnell employing histograms of local orientation. Their method uses the orientation histogram as a feature vector for gesture classification and interpolation. The method is simple and fast to compute, and it offers some robustness to scene illumination changes. However, this approach provides power and robustness at the expense of speed. In this method, a pattern recognition system is applied along with transformation T, which converts image or image sequence into a feature vector, which is then compared with the feature vectors of a training set of gestures. Euclidean distance metric is used for this purpose.

Another system for classification of hand postures against complex backgrounds in gray-level images has also been suggested by Triesch and von der Malsburg [7]. The presented system employs elastic graph matching which has already been successfully employed for face recognition. Elastic matching of a model graph M to an image means to search for a set XN of node positions simultaneously satisfying two constraints. The local image information attached to each node must match the image icon around the position where the node is placed. However, the distances between the matched node positions must not differ too much from the signal distances. The correct classification of this system is 86.2 % in comparison to others.

Naidoo et al. [8] are the members of another group of scholars who have jointly suggested a system that recognizes static hand images against complex backgrounds. In this method, a Support Vector recognition system is used to classify hand postures as gestures. Support Vector Machine (SVM) is a linear machine with some very good properties. The main idea of an SVM in the context of pattern classification is to construct a hyperplane as the decision surface. The hyperplane is constructed in a way to maximize the margin of separation between positive and negative examples. The SVM uses a coupled approach based on statistical learning theory known as natural risk minimization, which minimizes an upper bound on the minimization error (i.e., a heuristic parameter estimation seeks parameter values and minimizes the rate of "risk" or "loss" that the model on the training data has). Yet, another hand gesture recognition system has also been developed by Liesar and Sziranyi [9] based on the shape analysis of static gestures for HCI. This appearance-based recognition system uses modified Fourier descriptors (MFD) for the classification of hand shapes. This is also an example-based system which includes two phases, namely: training and running. In the training stage, the user shows the system one or more examples of hand gestures, and the system stores the carrier coefficients of the hand shape. In the running phase, the computer compares the current hand shape with each of the stored shapes by coefficients. The best gesture match is selected by the nearest-neighbor method with distance metric of MFD.

Symeonidis [2] has also presented an approach for static hand gesture recognition using orientation histograms as a simple and fast algorithm developed to work on a workstation. This approach recognizes static hand gestures, namely, a subset of American Sing Language (ASL). Pattern recognition system uses a transform that converts an image into vector, and then it compares the vector with the feature vectors of a set of gestures. The final system is implemented with a network. Hu and Meng [10] have developed a new visual gesture recognition method for the human–machine interface of mobile robot operation. The interface uses seven static hand gestures each of which represents an individual control command for the motion control of the remote robot, and adaptive object segmentation with color image in HLS representation is used and recognition is made by edge codes, matching, and skeletonizing.

A new approach has been presented by Chang and Chen [11] which recognizes static gestures based on Zernike moments (ZMs) and pseudo-Zernike moments (PZMs). This approach includes four stages. In the first step, an input static gesture is segmented into a binary hand silhouette via the modified color segmentation approach. In the second step, the binary hand silhouette is recommended with a Minimum Bounding Circle (MBC). In the third step, binary hand silhouette is decomposed into the finger part and palm part by morphological operations according to the radius of the MBC. The ZMs and PZMs of the important finger and palm parts are respectively computed based on the center of MBC. Finally, the nearest-neighbor techniques are used to perform the matching between an input feature vector and stored feature vector for static gesture identification. The proposed technique has recognition rate of 95 %.

Just [1] has proposed to apply an approach previously used for face recognition to the hand posture (static gesture) classification and recognition tasks. The feature is based on the local parametric pixel operator: modified Census transform (MCT) and information invariant. For the classification and recognition processes, a linear classifier is trained, using a set of feature lookup-benchmark in the field of posture recognition. Parvini and Shahabi [12] have jointly proposed an approach recognizing static and dynamic hand gestures by analyzing the raw streams generated by the sensors attached to human hands. The recognition of a sign is based on the observation of all forms of hand signs and finger-joint movements from a starting posture to a final posture. In this approach, the concept of the 'Range of Motion' is utilized for each joint movement. Range of Motion (ROM) is a quantity which defines the movement by measuring the angle from the starting position of the axis to its position at the end of the full range of the movement.

Radu et al. [13] have presented a real-time static-isolated gesture recognition application using a Hidden Markov Model approach with features extracted from gestures silhouettes. Nine different hand poses with various degrees of rotation are considered. The system, both simple and effective, uses color images of the hands to be recognized directly from the camera and is capable of processing 23 frames per second on a Quad Core Intel Processor. This work presented a fast, easy to implement solution to the static one hand gesture recognition problem. The system performance relies mostly on the skin segmentation block performance and recognition rate drops dramatically.

Another study proposes a method to recognize the number from 0 to 10. This method has three main steps: image capture, apply threshold, and recognizing the number. The study made an assumption that the user must wear color hand gloves [14].

1.1.1 Comparison with Other Approaches

Not many vision based approaches have been reported for static hand gestures considered in this work. The original contributions of this work are novel techniques to develop a vision-based gesture recognition algorithm to recognize a set of static hand gestures with scaling, translation, and rotation actions and movements with the idea of using two types of features: hand contour as a boundary-based feature in the first method and the complex moments (CMs) as a region-based feature in the second method. Unlike the earlier methods, we find that hand shape has not been explicitly considered as a possible feature. It was reported that use of hand shape makes it easier for the gesturer to remember commands [15]. In our study six static hand gestures in the proposed method; the proposed algorithm uses the multi-layer perception of neural network-based classifier. Gestural interfaces based on vision technologies are the most common way for the construction of advanced man–machine interfaces. However, in this method, the size of the image is used instead of the dedicated acquisition devices.

1.2 Scope of the Problem

Gestural interfaces based on vision technologies are the most way for the construction of advanced man–machine interfaces, but size of images instead of dedicated acquisition devices. There are three main problems: segmentation of the hand, tracking and recognition of the hand posture of (feature extraction and classification) [1].

(a) Segmentation
In most of the literature, hand segmentation has been performed using a controlled (uncluttered) background, known ground (i.e., background subtraction), segmentation by motion, and color segmentation (i.e., skin color filtering). Using controlled or backgrounds can be problematic in dynamic environments where backgrounds can change over time, and thus are non-realistic. Motion can be difficult to apply due to the artifacts of motion caused by light and camera motions. Color segmentation is a fast and robust approach to hand segmentation that works well under lighting conditions and against unknown backgrounds. It cannot be used if the background behind the hand is close to the color of the segmentation.

(b) Tracking
Articulated objects (such as the hand) are more difficult to track than the rigid objects. The analysis process of the human hand is further articulated by the fact that the hand is a non-rigid articulated structure and changes in shape in various ways. The four fingers are adjacent to other that leads to self-occlusions. To overcome these difficulties, tracking has been facilitated through the use of special markers (colored gloves or color marked gloves) and a combination of color and shape constraints. Tracking can be done using on single camera (mono) or using multiple cameras (stereo). To track the hand in 3D, a model of the hand is needed. Hand tracking needs to update well-chosen parameters through consecutive images; and this problem is strongly tied with hand segmentation in each image alone the tracking process [1].

(c) Recognition
The major difficulties of hand posture (static gesture) and gesture recognition are feature extraction and the recognition itself. Because hand postures and gestures are highly variable from one person to another, it is essential to capture their essence—their invariant properties—and use this information to represent them. The features must optimally distinguish the variety of hand gestures or postures from each other and make recognition of similar gestures of postures simpler. Another problem related to hand gestures recognition is gesture segmentation (or gesture spotting). Hand gesture spotting consists of determining the start and end points of a particular gesture. Gesture spotting needs also to deal with the rejection of unknown gestures.

1.3 The Basics of Gesture Recognition

The general gesture recognition process in systems of any type can be broken down into the following components as shown in Fig. 1 [16].

The first stage, as displayed in the figure, is mostly concerned with the hardware of the system and the way data for the recognition process is gathered (in the form of bitmaps or lists of vertices). The second stage is a preprocessing stage. In this stage, edge-detection, as well as smoothing and other filtering processes, can occur. This stage prepares the data for the main computational stage for feature extraction. Some systems might never use the term ‘feature’, but somewhere along the line they will find a way of quantifying their input. The features of the input are then evaluated in one or more of several possible ways to make a decision about which gesture the system is most likely subjected to in the fourth stage, also known as evaluation stage. Nevertheless, all systems will

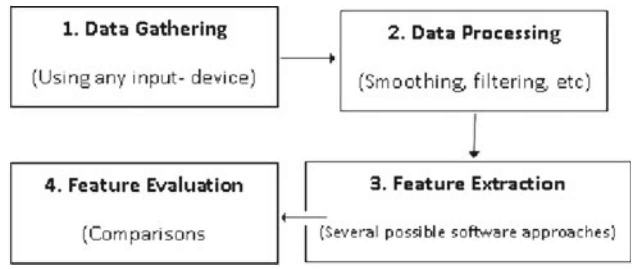


Fig. 1 Overview of gesture recognition system

have a limited set of gestures that they can recognize at any given time [16].

1.4 Preprocessing

1.4.1 Segmentation

Segmentation is the initial stage for any recognition process whereby the acquired image is broken up into meaningful regions or segments. The segmentation process is not primarily concerned with what the regions represent but only with the process of partitioning the image. In the simples case (binary images) there are only two regions: a foreground (object) region and a background region. In gray-level images there may be many types of region or classes within the image for example in a natural scene to be segmented there may be regions of clouds, ground, building, and trees [17]. Segmentation subdivides image into its constituent parts of objects; the level to which this subdivision is carried depends on the problem being solved. That is segmentation should stop when the objects of interest in an application have been isolated [19]. There are two main approaches to segmentation:

1. Pixel-based or local methods that includes edge detection and boundary detection.
2. Region-based or global approaches, which include region merging and splitting and threshold [17].

Thresholding the simple’s image segmentation problem occurs when and image contains an object having homogeneous intensity and a background with different intensity levels [21]. Thresholding techniques are used to partition the image histogram using a single threshold, T. segmentation is then accomplished by scanning the image pixel by pixel and labeling each pixel as object or background depending on whether the gray level of that pixel is greater of less than the value of T [19].

1.4.2 Noise Reduction

Spatial filters can be effectively used to remove various types of noise in digital images. These spatial filters typically operate on small neighborhoods, 3×3 to 11×11. Many spatial filters are implemented with convolution masks. Because a convolution mask operation provides a result that is a weighted sum of the values of a pixel and its neighbors, it is called a linear filter. The mean filters are essentially averaging filters; they operate on local groups of pixels called neighborhoods and replace the center pixel with an average of the pixels in this neighborhood. This replacement is done with a convolution mask [18]. The median filter is a nonlinear filter. A nonlinear filter has a result that cannot be found by a weighted sum of the neighborhood pixels as done with convolution mask. However, the median filter does operate on a local neighborhood; after the size of the local neighborhood is defined the center pixel is replaced with the median or center value present among its neighbors rather than by average [18]. The median filter disregards extreme values (high or low) and does not allow them to influence the selection of a pixel value which is truly representative of the neighborhood. It is therefore good at removing isolated extreme noise pixels (often known as 'salt' and 'pepper' noise), while substantially retaining spatial detail. However, its performance deteriorates when the number of nose pixels is more than half the number of pixels in the window [17].

1.4.3 Edge Detection

Edges are basic image features. They carry useful information about object boundaries which can be used for image analysis object identification and for image filtering applications as well [21]. Edge detection methods are used as a first step in the line detection process and also used to find complex object boundaries by marking potential edge points corresponding to places in an image where changes in brightness occur. After these edge points have been marked they can be merged to form lines and object outlines. Edge detection operations are based on the idea that edge information in an image is found by looking at the relationship a pixel has with its neighbors with widely varying gray levels which may represent an edge point. In other words an edge is defined by a discontinuity in gray-level values. Ideally, an edge is caused by changes in color or texture or by the specific lighting conditions present during the image acquisition process [18].

(a) Sobel Operator
Sobel operator is recognized as one of the best 'simple' edge operators it utilizes two 3×3 masks [17]. The Sobel edge detection masks look for the horizontal and vertical directions and then combine this information into a single metric. They are as follows (Fig. 2):
These masks are each convolved with the image. At each pixel location there are two numbers: S1 corresponding to the result from the row mask and S2 from the column mask these numbers are used to compute two metrics, the edge magnitude and the direction which are defined as follows [18]

-1	-2	-2
0	0	0
1	2	1

Row Mask

-1	0	1
2	0	2
-1	0	1

Column Mask

Fig. 2 Sobel operator

-1	-1	-1
0	0	0
1	1	1

Row Mask

-1	0	1
1	0	1
-1	0	1

Column Mask

Fig. 3 Prewitt operator

$$\text{EdgeMagnitude} = \sqrt{S1^2 + S2^2} \tag{1}$$

$$\text{EdgeDirection} = \tan^{-1}\left(\frac{S1}{S2}\right) \tag{2}$$

(b) Prewitt Operator
Prewitt operator is similar to the Sobel operator but with different mask coefficients, the masks are defined as shown in (Fig.3). The edge magnitude and the direction are defined as follows [18].

$$\text{EdgeMagnitude} = \sqrt{P1^2 + P2^2} \tag{3}$$

$$\text{EdgeDirection} = \tan^{-1}\left(\frac{P1}{P2}\right) \tag{4}$$

(c) Laplacian Operator
The Laplacian operator is a second-order derivative operation that has zero crossing (e.g., transition from positive to negative and vice versa) [19,21]. Laplacian masks are rotationally symmetric, which means edges at all orientations contribute to result. They are applied by selecting one mask and convolving it with the image. The sign of the result (positive or negative) from two adjacent pixel locations provides directional information and tells as which since of the edge is brighter [18].

1.4.4 Coordinate Normalization

The idea of coordinate normalization is to map the scaled hand image coordinates to the standard size of range $[-1, +1]$ [28]. The purpose of this step is to keep domain of image coordinates fixed and irrelevant to the original size. The condition of keeping the domain of the coordinates within the limited boundaries will effectively satisfy the convergence of higher ordered moments. Thus, the scaled image coordinates (x, y) will be transformed into the normalized set (x, y), which can be considered as a "standard" version of the original coordinated (x, y). Using centre of image, each pixel coordinate (x_s, y_s) value is mapped to the domain $[-\mathbf{1}, +1]$, which can be done using the following equations:

$$X_n = \left(\frac{2}{W-1} \times X\right) - 1 \tag{5}$$

$$Y_n = \left(\frac{2}{H-1} \times Y\right) - 1 \tag{6}$$

where H and W are the height and width of the scaled image respectively.

1.5 Feature Extraction

Feature extraction is part of the data reduction process and is followed by feature analysis. One of the important aspects of feature analysis is to determine exactly which features are important [18]. Feature extraction is a complex problem and often the whole image or transformed image is taken as input. The goal of feature extraction is to find the most discriminate information in the recorded images. Feature extraction operates on two-dimensional image array but produces a list of descriptions or a 'feature vector' [17,22]. Mathematically, a feature is an n-dimensional vector with its components computed by some image analysis. The most commonly used visual cues are color, texture, shape, spatial information, and motion in video. For example, the color may represent the color information in an image such as color histogram color binary sets, color coherent vectors. The n components of a feature may be derived from one visual cue or from composite cues such as the combination of color and texture [20].

Selecting good features is crucial to gesture recognition since hand gestures are very rich in shape variation, motion, and textures. For static hand gesture recognition, although it is possible to recognize hand posture by extracting some geometric features such as fingertips, finger directions, and hand contours, such features are not always available and reliable due to self-occlusion and lighting conditions. There are also many other non-geometric features such as color, silhouette, and textures, however, they are inadequate in recognition. Since it is not easy to specify features explicitly, the whole image or transformed image is taken as the input and features are selected implicitly and automatically by the recognizer [4].

-5	-6	-5.5	-5	-5.5	-6	-5
-6	-5	-2	0.5	-2	-5	-6
-5.5	-2	0.4	0.4	0.4	-2	-5.5
-5	0.5	0.4	225	0.4	0.5	-5
-5.5	-2	0.4	0.4	0.4	-2	-5.5
-6	-5	-2	0.5	-2	-5	-6
-5	-6	-5.5	-5	-5.5	-6	-5

Fig. 4 A 7 × 7 surround mask

1.5.1 Contour Detection

Contour detection in real images is a fundamental problem in many computer vision tasks. Contours are distinguished from edges as follows. Edges are variations in intensity level in a gray-level image whereas contour is salient coarse edges that belong to objects and region boundaries in the image. By salient is meant that the contour map drawn by human observers includes these edges as they are considered to be salient. However, the contours produced by different humans for a given image are not identical when the images are of complex natural [23]. The usefulness of contour detection in a great number of applications has been well established and demonstrated. Indeed, this operation is of great help for further image analysis and scene understanding. There are many kinds of edge detectors most of them are based on derivative operators that give a high response at the contour points and a low response in homogeneous areas. The oldest and simplest edge detector is undoubtedly the digital gradient operator [24]. Contour detection can be implemented in a simple manner as follows:

1. Compute a gradient map: this gradient computation must performed in two orthogonal directions using Sobel mask.
2. Incorporate the surround influence on the gradient map: this can be implemented as a convolution operation with an appropriate isotropic mask as shown in Fig. 4.
3. Binaries the output of the second stage using a standard procedure [23].

1.5.2 Invariant Features

Features associated with images are called 'invariant' if they are not affected by certain changes regarding the object view point. It is widely accepted that invariant features would be independent of modifiers such as translation, scaling, rotation, and light conditions. Ideally invariant features should recognize objects whose geometry can change because either the object moving in relation to the camera is articulated or different viewpoints cause different patterns in 2D image. Usually these modifiers are not independent of each other and therefore they often happen simultaneously. It is also agreed that there is not truly pure invariant feature. Rather there are features that are more or less robust to none or more modifiers [25]. The need for invariant features rises in many practical problems as illustrated in the following example.

1. *Speech Recognition* the adopted features should be independent of speakers. Hence, features should be insensitive to speaker-dependent data.
2. *Speaker Recognition* features should be invariant to spoken speech.
3. *Image Recognition* features should be invariant under rotation, translation, scaling, and the illumination angle used in generating the image [26].

1.5.3 Complex Moments

The notation of complex moment was introduced by Abo-Mostafa [27] as a simple and straightforward way to derive moment invariants. The complex moment of order m is defined as:

$$C_m = \iint (x + iy)^m \mu(x, y)\, \mathrm{d}x\, \mathrm{d}y \tag{7}$$

where $i = \sqrt{-i}$ and $\mu(x, y)$ is the real image intensity function.

The complex moments have been proposed as a solution to different pattern recognition problems. The CMs are very simple and quite powerful in providing an analytic characteristic for moments invariant. A Moment invariant is a set of moment values extracted from the image data in such a way that their values are invariant to the rotation of the image data. Moreover, the value of a CM could be considered a moment invariant if that value can be computed from a group of CMs for the same object at different resolution [28]. Moment invariant can be used as a feature of classification and recognition of an object. Complex moments have two parts: real part and imaginary part; however, the computation of their values decomposes into two directions: x-axis moment which represents real part direction and y-axis moment for the imaginary part direction [28]. It is well known that the moment sets can offer a powerful description on the geometrical distribution of the material within any region of interest. The low order of complex moments has meanings very relevant to some well-known physical quantities [28].

Zero-order moments represent the total mass of the image. 1st-order moments together with zero-order moments assign the center of the mass of the image. 2nd-order moments represent moment of inertia. 3rd-order and 4th-order moments are used for computing statistical quantities known as skews and kurtosis, respectively. While the higher nth-order moments give additional statistical and structural information about the image. The computation for complex moment should involve the calculation of its real and imaginary components and then, nth-order complex moment M_i. For the hand image of size $n \times m$ is calculated according to the following equation:

$$M_i = \sum_{x=0}^{n-1} \sum_{y=0}^{m-1} Z_i^n a(x, y) \tag{8}$$

where i indicates moment's order, $Z_n = X_n + iY_n$ is a complex number, $a(x, y)$ represent a pixel value at the position, (x, y) may be ON or OFF pixel. The calculation of complex moments may require long computation time, to reduce this relation below will be used:

$$Z^n = Z^{n-1} \cdot Z \tag{9}$$

where $Z = x + iy$ is the complex number when the real and imaginary parts of Z^n and Z^{n-1} are assumed to be complex numbers that could be written as follows:

$$Z^n = R_n + iI_n \tag{10}$$

and

$$Z^{n-1} = R_{n-1} + iI_{n-1}. \tag{11}$$

By taking into consideration the case Z^0 and Z^1, it is simple to calculate that $R_0 = 1$; $I_o = 0$; $R_1 = X$; $I_{o1} = Y$; substituting the value of Z_n, Z_{n-1} and Z_1 in Eq. 7 yields:

$$R_1 = R_{n-1}X - I_{n-1}Y \tag{12}$$

$$I_1 = l_{n-1}X - R_{n-1}Y \tag{13}$$

These equations indicate that knowing the components of Z^{n-1} will be directly used to compute the components of Z^n [28].

1.5.4 Artificial Neural Networks

An artificial neural network (ANN) is an information processing system that has certain performance characteristics in common with biological neural networks. Artificial neural networks have been developed as generalizations of mathematical models of human cognition or neural biology based on the assumptions that

1. Information processing occurs at many simple elements called neurons.
2. Signals are passed between neurons over connection links.

3. Each connection link has associated weight, which in a typical neural net, multiplies the signal transmitted.
4. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) [29,30].

Artificial neural networks are widely recognized as powerful classification methods that have been used extensively to solve many problems in different domains [31,32]. In the current study, multilayer perceptron (MLP) trained with back-propagation learning method [30] which is the most popular supervised neural networks model is selected as a classifier to recognize the static hand gesture.

2 Methodology

This section will introduce the proposed vision-based static hand gesture recognition algorithm to recognized the chosen set of six static hand gestures as commands used for HCI, these gestures are defined as: Open, Close, Cut, Paste, Maximize, Minimize. The proposed algorithm consists of three basic stages: pre-processing feature extraction, and classification. As known the main problems of 2D object recognition are the size changing, translation of position, and rotation by angle from the principle axes. In this algorithm two methods are proposed; in the first method the hand contour is extracted as a geometric feature and the method treats the problems of size changing and translation (in some cases). The second method treats the problem of rotation in addition to previous problems using the hand complex moments feature. The extracted features are entered to the last stage (neural networks using the supervised back-propagation learning algorithm), and this stage is responsible for recognizing and deciding to which class the hand gesture belongs. The main efforts in this work and main contribution are proposed in the feature extraction and classification parts. It is known that to have a robust classification it is necessary to choose properly the features as well as the correct way to present the features to the classifier. This work explores mainly this topic finding proper features and proper way to represent them that matches the classifier characteristics.

The proposed hand gesture recognition algorithm consists of two methods: neural network with hand contour and neural network with hand complex moments. Either of these methods is formed by four sequential stages: hand image capture, image preprocessing, feature extraction, and classification, as seen in Fig. 5.

The role of the preprocessing module is to segment the pattern of interest (hand) from the background, removing noise and any other operation which will contribute to defining a compact representation of the pattern. In the training phase, the feature extraction module finds the appropriate features for representing the input patterns and the classifier is trained using the back-propagation algorithm. In the testing phase, the trained classifier will assign the input pattern (test hand gesture image) to one of the pattern classes under consideration based on the measured features. The next sections present a detailed explanation of the proposed methods.

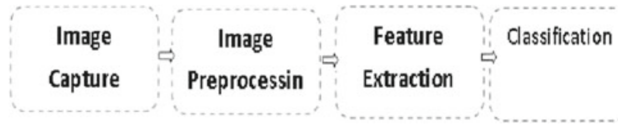


Fig. 5 Overview of gesture recognition system

2.1 Hand Gestures Image Database

The starting point of this study was the creation of a database with all the hand gesture images that would be used for training and testing. The construction of such a database is clearly dependent on the application. Each gesture represents a gesture command mode. These commands are widely used in various application programs. Therefore, these gestures are allowed to be flexible and natural so that they can successfully be applied to a wide range of people and situations. The gestures images are real images with different sizes acquired using digital camera and taken from one subject. The database consists of 30 images for training set, with five samples for each gesture, and 84 images derived from the two remaining samples of 6 gesture classes with scaling effect, translation, and rotation.

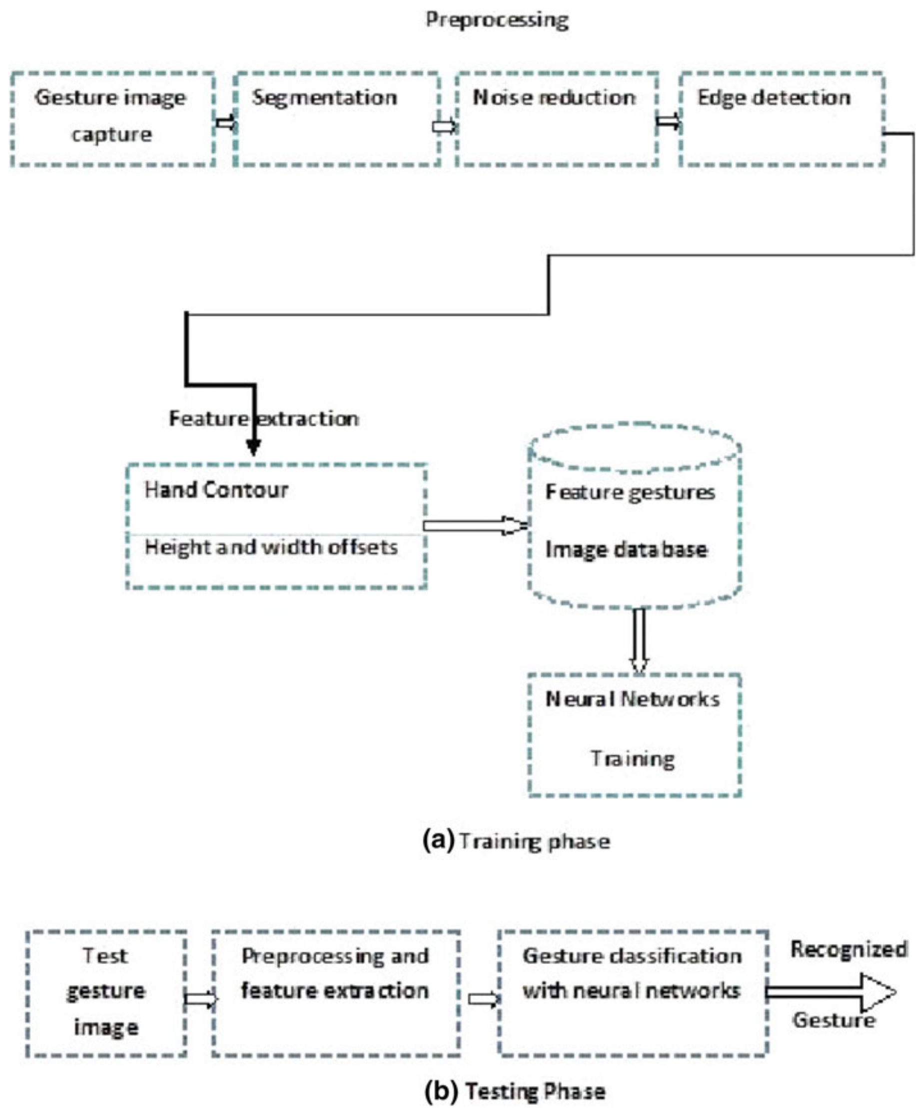
2.2 Neural Network with Hand Contour

As shown in Fig. 6, the recognition process consists of two phases: training (learning) and classification (testing). Steps included in each phase are as described in the following lines.

A. Training Phase

(1) Capturing Gesture Image from training set.
(2) Segmentation to isolate hand object from the background.
(3) Noise reduction to remove any noise from the segmented image.
(4) Edge detection to find the hand gesture image boundaries.
(5) Contour detection as a geometric feature.
(6) Calculating the height offset and width offset as a general feature.
(7) Saving feature image as a training pattern.

Fig. 6 Block diagram of the proposed static hand gesture recognition algorithm; **a** training phase, **b** testing phase



(8) Learning using neural networks with back-propagation algorithm.

B. Testing Phase

(1) Capturing Gesture Image from testing set.
(2) Segmentation to isolate hand object from the background.
(3) Noise reduction to remove any noise from the segmented image.
(4) Edge detection to find the gesture image boundaries.
(5) Contour detection as a geometric feature.
(6) Calculating height offset and width offset as a general feature.
(7) Recognized by neural networks.

2.2.1 Preprocessing

The primary goal of the preprocessing stage is to ensure a uniform input to the classification network. This stage includes hand segmentation to isolate the foreground (hand gesture) from the background, removing any noises caused by segmentation process using special filters. This stage also includes edge detection to find the final shape of the hand.

(a) Hand segmentation
The hand image is segmented from the background. The segmentation process, however, should be fast, reliable, and consistent, and able to produce the best-quality output possible giving the constraints, which must produce an image suitable to recognize the gesture of the hand. In this work, a Thresholding algorithm is used for segmentation of the gesture image as it is fast and requires a very low time comparing with the other methods which

is a very important feature especially for real-life applications.

(b) Noise reduction
Once the hand gesture image has been segmented, a special filter is applied. The goal of applying a filter is to eliminate all the single white pixels on a black background and all the single black pixels on a white foreground. To accomplish this goal, a median filter is applied to the segmented image.

(c) Edge detection
For recognizing static gestures in our system, the model parameters are derived from description of the shape, and the boundary of the hand is extracted for further processing. Therefore, several different edge-detector operators were tried including Sobel and Prewitt. Another attempt was using a Laplacian edge detector which gives thin edges. Finally, Sobel has been chosen for its very good results.

2.2.2 Gesture Feature Extraction

The objective of the feature extraction stage is to capture and discriminate the most relevant characteristics of the hand gesture image for recognition. The selection of good features can strongly affect the classification performance and reduce the computational time. These selected features, consequently, result in an easier classification task. The features used must be suitable for the application and the applied classifier. In a proposed method two types of features are extracted, namely, the hand contour as a geometric feature and height width offsets of the hand image as assistant general features.

(a) Geometric Feature (Hand Contour)
One of the most important geometric features for static hand gesture recognition is hand contour. Figure 7 shows the pseudo code for contour detection algorithm. Once the contour map is produced, feature image scaling takes place. Feature image scaling is a very simple method which reduces the feature image size by selecting several rows and columns. For all the images in the training set, a scaling is performed. The result is feature image with 32, the number of the rows, and 32, the number of the columns.
After the feature image scaling, a feature gesture image must be prepared before entering the last stage, i.e., classifier stage. The preparing step includes shifting the hand gesture section to the origin point (0, 0) to solve translation problem.

(b) General Features
General features describe the hand image offsets that serve as an additional information such as height and width of the hand image. The general features will be represented as a 6 × 6 matrix; its indices are 2 to power (0, 1, 2, 3, 4, 5) as shown in Fig. 8.
This matrix will hold values of two (ones) with all remaining zero values. The two (ones) positions represent height and width of the hand gesture. For example as shown in the Fig. 8 if the hand gesture has a width equal to 8 or near to this value (using threshold value), and the height equal to 32 or near to this value, the position of 2^3 will hold a (one) value, the same thing for the height the position of 2^5 will hold a (one) value. This feature matrix is compounded with contour feature vector as composed features to produce a new feature vector with 1,060 elements, then it is passed to the back-propagation neural network as its input vector.

```
Input: Gesture Image has passed the Segmentation and Edge Detection Process
Output: A Gesture Image with Contour Detection
FOR RM=1 ToMaskWidth
FOR CM= 1 ToMaskHeight
FULL Contour\_Mask
END FOR
END FOR
FOR each pixel in the Gesture Image
CONVOLUTE Gradients from Edge Detector with Contour\_Mask
RETURN Surround\_Influence\_Imge
END FOR
FOR each pixel in the Surround\_Influence\_Image
IF pixel value$>$ ``specific threshold'' THEN
Mark each pixel in the Surround\_Influence\_Image as Contour
ELSE
Mark it as lack Contour.
END IF
END FOR
END
```

Fig. 7 The pseudo code for contour detection

	2^0	2^1	2^2	2^3	2^4	2^5
2^0	0	0	0	0	0	0
2^1	0	0	0	0	0	0
2^2	0	0	0	0	0	0
2^3	0	0	0	0	0	1
2^4	0	0	0	0	0	0
2^5	0	0	0	1	0	0

Fig. 8 Example of general futures 6 × 6 matrix

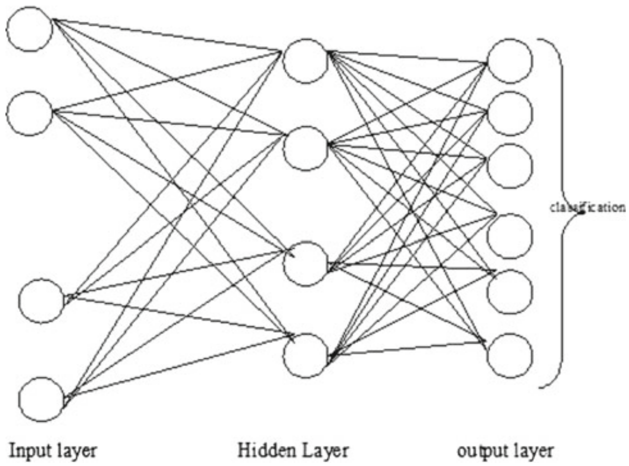


Fig. 9 Gesture recognition network architecture

Table 1 Parameters for the five multilayer neural networks

Parameters	Values
Input layer	1,060 nodes
Hidden layer	100 nodes
Output layer	6 nodes
Stop error	0.01
Learning rate	0.9

2.2.3 Neural Network-Based Classifier

The last stage of the proposed system is the classification. The right choice of the classification algorithm to be used in a gesture recognition system is highly dependent on the properties and the format of the features that represent the gesture image. In this work a standard back-propagation neural network is used to classify gestures. The network consists of three layers; the first layer consists of neurons that are responsible for inputting a hand gesture sample into the neural network. The second layer is a hidden layer. This layer allows neural network to perform the error reduction necessary to successfully achieve the desired output. The final layer is the output layer with one node per class. Typically the number of neurons in this layer is determined by the size of the set of desired outputs, with each possible output being represented by a separate neuron. The structure of this particular back-propagation neural network is illustrated in Fig. 9.

There are six outputs from the neural networks, each output represents index for one of the six hand gesture images classes. The highest index value (in the testing phase) will represent the recognized gesture image. For the recognition process, five neural networks with same structure are used. The parameters for the multilayer neural networks are shown in Table 1.

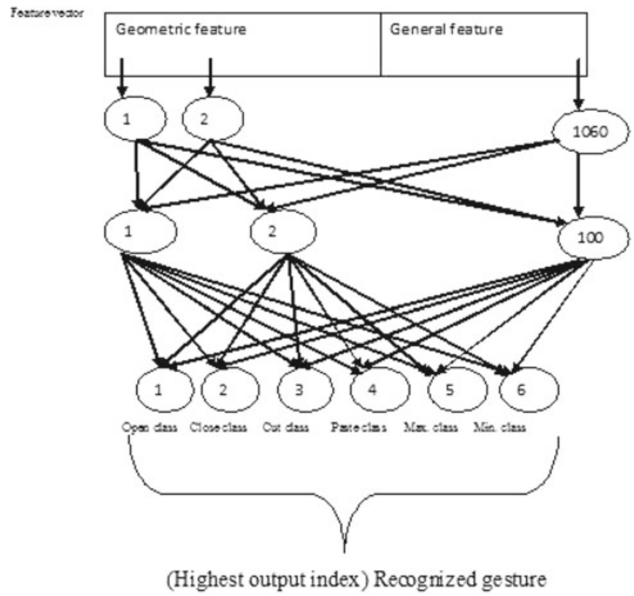


Fig. 10 Detailed design of neural network

In back-propagation, a hand gesture sample is propagated through the multilayer neural network, producing an output. This output is compared to the desired output, giving an error rate for the output layer. Because the error rate of a neuron is a function of the error rates of all the units that use its output, the error rates of the layer directly below the output layer can now be found. These error rate calculations will continue to propagate through the network in a backward fashion, until the error rates for all the neurons have been found. Each neuron will then make slight weight adjustments in order to minimize its error signal. This pattern is repeated until the error rate of the output layer reaches a minimum value. This process is then repeated for the next input value, until all of the input values have been processed the detailed design of the classification stage is ilustrated in (Fig. 10).

2.3 Neural Networks with Complex Moments

Figure 11 demonstrates the steps of preprocessing operations. The sample for each hand gesture is trimmed by clipping the body of the hand gesture for empty columns and lines, normalizing the size of the clipped gesture to standard size. In a feature extraction stage, the complex moment value is extracted.

2.3.1 Preprocessing

In addition to segmentation and noise reduction processes as in previous method, the preprocessing involves another operations that includes image trimming, scaling, and coordinate normalization.

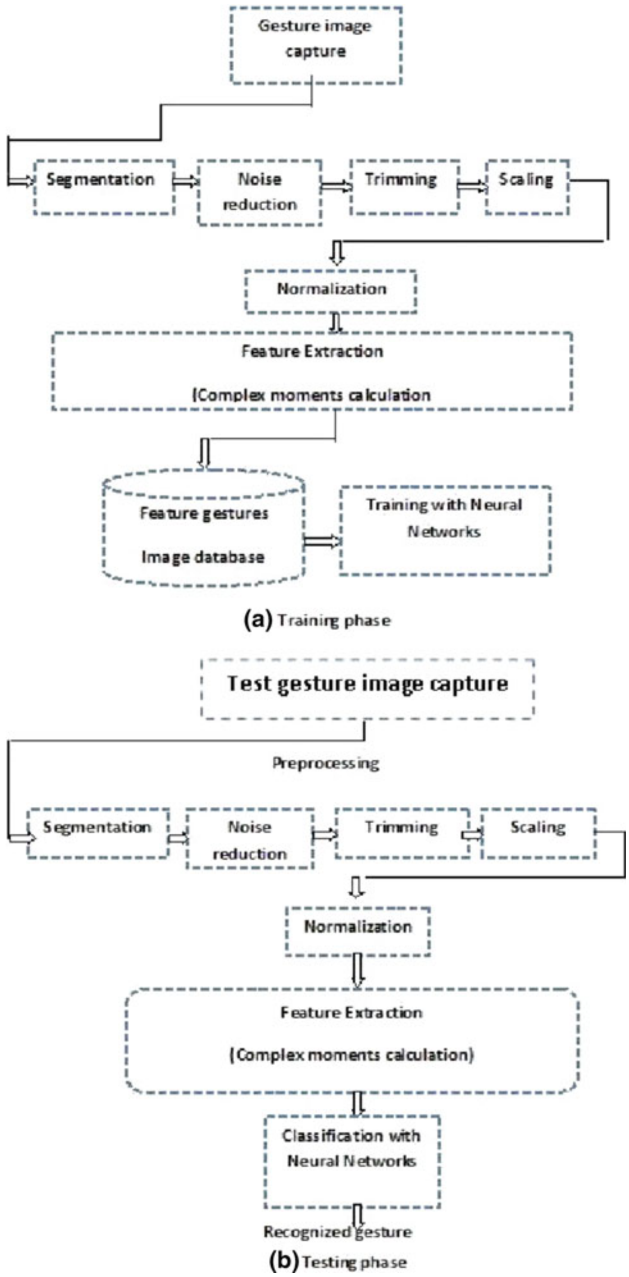


Fig. 11 Block diagram of the second proposed method for static hand gesture recognition algorithm. **a** Training phase, **b** testing phase

(a) Image Trimming
The image of hand gesture may include additional empty lines and columns that have no data (space lines); these empty lines should be eliminated by tracing from outside margins towards inside and stopping at the first occurrence of the ON pixel at each side of the four edges.

(b) Image Scaling
The dimensions of the hand gesture images are varying due to capturing process therefore the image size is adjusted to fixed size (250 × 250) in order to facilitate the calculation of the next stage.

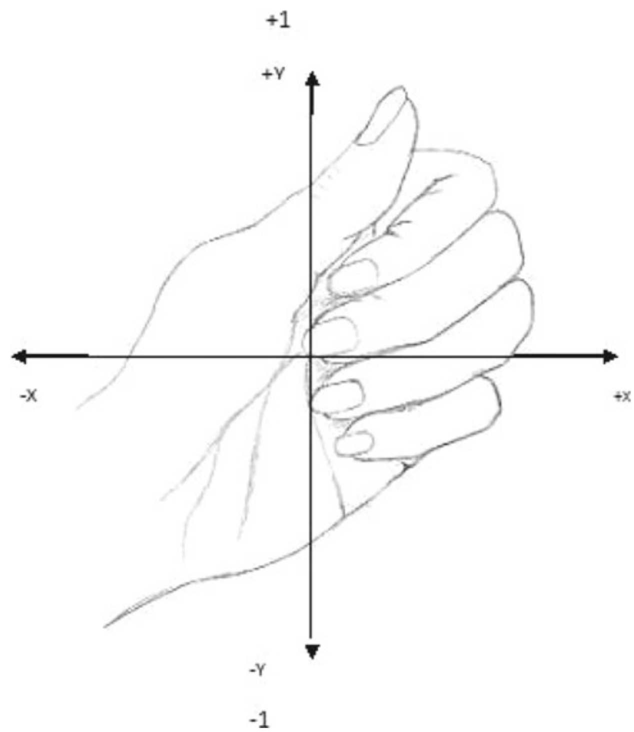


Fig. 12 Central coordinates normalization

(c) Coordinate Normalization
After image scaling the coordinate normalization process is taken place to map the scaled hand image coordinates to the standard size ranging between $[-1, +1]$ using the equations in previous Sect. 1.4.4. The origin point coordinates will be at the center of the image as shown in Fig. 12.

2.3.2 Feature Extraction

In this proposed method for gesture recognition the complex moments are used as a moment invariant which is a set of moment values extracted from the image data that their values are invariant to scaling translation and rotation of the image data. Feature extraction stage includes the calculation of complex moments. For every hand gesture in the database, these features are calculated.

(a) Complex Moments Calculation
For each hand gesture image, the complex moments from zero- to ninth-order will be calculated (using the equations in Sect. 1.4.4) so each feature vector will have ten values, as shown in Fig. 13. With many experiments it found that this (ten values) is enough to represent each hand gesture and to make it recognizable. In order to make the complex moments values in a small range before they enter the neural network classifier, they

M_0	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9

Fig. 13 The complex moments feature vector

```
Input: Normalized Gesture Image
Output: Feature vector with (n) complex moments values
FOR MomOrder= 0 ToMomentNumber
SET Mom(MomentOrder).RealPart=0
SET Mom(MomentOrder).ImgPart=0
END FOR
SET Real0=1
SET Imgn0=0
FOR Row=0 TO ImagHeight-1
FOR Col=0 TO ImageWidth-1
Mom(0).RealPart =Mom(0).RealPart + Array (Row,Col)
END FOR
END FOR
FOR MomOrder=1 ToMomentNumber
FOR Row=0 TO ImageHeight-1
FOR Col=0 TO ImageWidth-1
RealT=Real0*Row-Img0*Col
ImgT=Img0*Row+Real0*Col
Mom(MomOrder).RealPart=Mom(MomOrder).RealPart
+RealT*Array(Row,Col)
Mom(MomOrder).ImgPart=(MomOrder).ImgPart
+ImgT*Array(Row,Col)
END FOR
END FOR
MomValue(MomOrder)=SQRT(SQR(MomOrder).RealPart
+SQR(Mom(MomOrder).ImgPart))
Real0=RealT
Img0=ImgT
END FOR
```

Fig. 14 Complex moments calculation algorithm

should be scaled to small range [0, 1]; this will facilitate the classification process.

(b) Complex Moments Algorithm
The computation for complex moments illustrated in the pseudo code is as shown in Fig. 14.

2.3.3 *Neural Network-Based Classifier*

As in previous method, a multilayer neural network (with the same structure) is used for recognizing the hand gesture. The input of this network is a feature vector with ten values; the parameters of this neural network are shown in Table 2. The remaining details about recognition process is the same as in Sect. 2.2.3.

3 Experiments and Results

Hand gesture recognition is a decision concerning the gesture of the user which cannot be made unless the gesture or recognition algorithm passes through several operations on unknown hand gesture. These operations are required before deciding to which class the gesture belongs. Hand gesture recognition algorithm includes two methods, extracting the hand contour as a geometric feature and treating the problem of rotation in addition to previous problems using the hand complex moments feature. Each of these methods consists of two phases: training phase and testing phase. The training phase works on the gesture used to train neural networks while the testing phase works on the hand gestures used to decide to which class they belong, which will not be used in the training phase. Both phases consist of two main processes: preprocessing and feature extraction. In this section, the results obtained from applying the proposed hand gesture recognition algorithm to both methods and the effects of each of the processes are presented. For each hand gesture of the six selected gesture classes, seven samples are taken under different light conditions (natural and artificial light conditions). In the following sections the effect and the performance of the algorithm will be presented and discussed.

Table 2 Parameters for the five multilayer neural networks

Parameters	Values
Input layer	10 nodes
Hidden layer	100 nodes
Output Layer	6 nodes
Stop error	0.01
Learning rate	0.9

3.1 Contour with ANN

3.1.1 *Preprocessing Stage Effects*

In this stage many processes are performed on the hand gesture image to make it ready for the following feature extraction stage. The effects of these operations will be explained in the following subsections.

(a) Hand Gesture Segmentation
All the techniques that are used in this paper are based on hand shape. The acquiring color gesture image is segmented to isolate the foreground hand from the background.

(b) Noise Reduction
The segmented hand image may contain some noises that will affect the result values produced by the feature extraction stage. Hence, the values of the computed features will be different for the same image if it contains noise. So using the median filter will reduce the noise as much as possible.

Table 3 Parameters for the five multilayer neural networks

Parameters	Values
Input layer	1,060 nodes
Hidden layer	100 nodes
Output layer	6 nodes
No.of images for each training set	6
Stop error	0.01
Learning rate	0.9

(c) Edge Detection
The Edge detection process is performed using Sobel operator objects presented in an image.

3.1.2 Feature Extraction

The hand gesture image that has passed through image preprocessing stage is fed to the feature extraction stage to compute the feature information about the image. As soon as contour feature is computed the image size is adjusted so that each hand gesture image size becomes 32×32. Image resizing makes the system faster; this operation will reduce the negative effects of the size change. The general features (height offset and width offset) will be computed implicitly.

3.1.3 Artificial Neural Networks Training and Testing Phases

1. Training Phase
In this phase, the composite feature vectors computed earlier and stored in a feature images database are fed to the next stage of our system as inputs. These feature vectors are used to train the neural networks. The learning process for the five multilayer neural networks is accomplished using the following parameters as shown in Table 3. The neural networks are trained through successive epochs (iteration), after each epoch the square error over the validation set is computed. The training results, as shown in Fig. 15, show the convergence of learning algorithm for back-propagation neural networks and the learned samples with respect to the number of epochs for each network.
2. Testing Phase
After training neural networks, performance is estimated by applying the testing set to the network inputs and computing the classification errors. The used activation function is binary-sigmoid which holds outputs constantly between 0 and 1. Then, the test gesture feature image will be entered into the first neural network. In this phase, if the network succeeds to recognize the gesture, the test operation is stopped. If this network does not recognize the gesture features, the second network will be activated and so on. If all networks fail to identify the features, "gesture not recognized" message will appear to announce the failure in recognition. In this phase, 56 hand gesture images are used to test the system with different light conditions and with scaling and translation effects. The system now is able to recognize and classify any unknown gestures if they are in the original database. Each gesture has a table of recognition results and with neural network outputs for one gesture image the performance of the proposed system is evaluated based on its ability to correctly recognized gestures to their corresponding input gestures, the metric that is used to accomplish this job is called the recognition rate. The recognition rate is defined as the ratio of the number of correctly recognized gestures to the total number of input gestures as shown in the equation:

$$\text{Recognition rate} = \frac{\text{No of correctly recognized gestures}}{\text{Total number of input gestures}} \times 100 \tag{14}$$

A summary of all recognition results and the recognition rates for each of the six hand gestures is presented by Table 4, and the recognition rates with each class are shown in Fig. 16.

3.1.4 Complex Moments with ANN

The preprocessing stage in this method includes image trimming followed by normalization process. The effect of these operations will be presented in the next sections.

3.1.5 Preprocessing Stage Effect

(a) Image Trimming Effect
The hand gesture filtered image may contain unused space surrounding the hand gesture so the image trimming process is used to extract the hand gesture from its background.
(b) Coordinate Normalization
After scaling each image size to fixed size 250×250, the coordinates for the hand image are normalized between $[-1, +1]$.
Each hand gesture in the training set will have a feature vector of ten values and these values represent the complex moments starting with zero order up to nine order. Tables 5 and 6 present an example of the results of these computations, before and after normalization process.

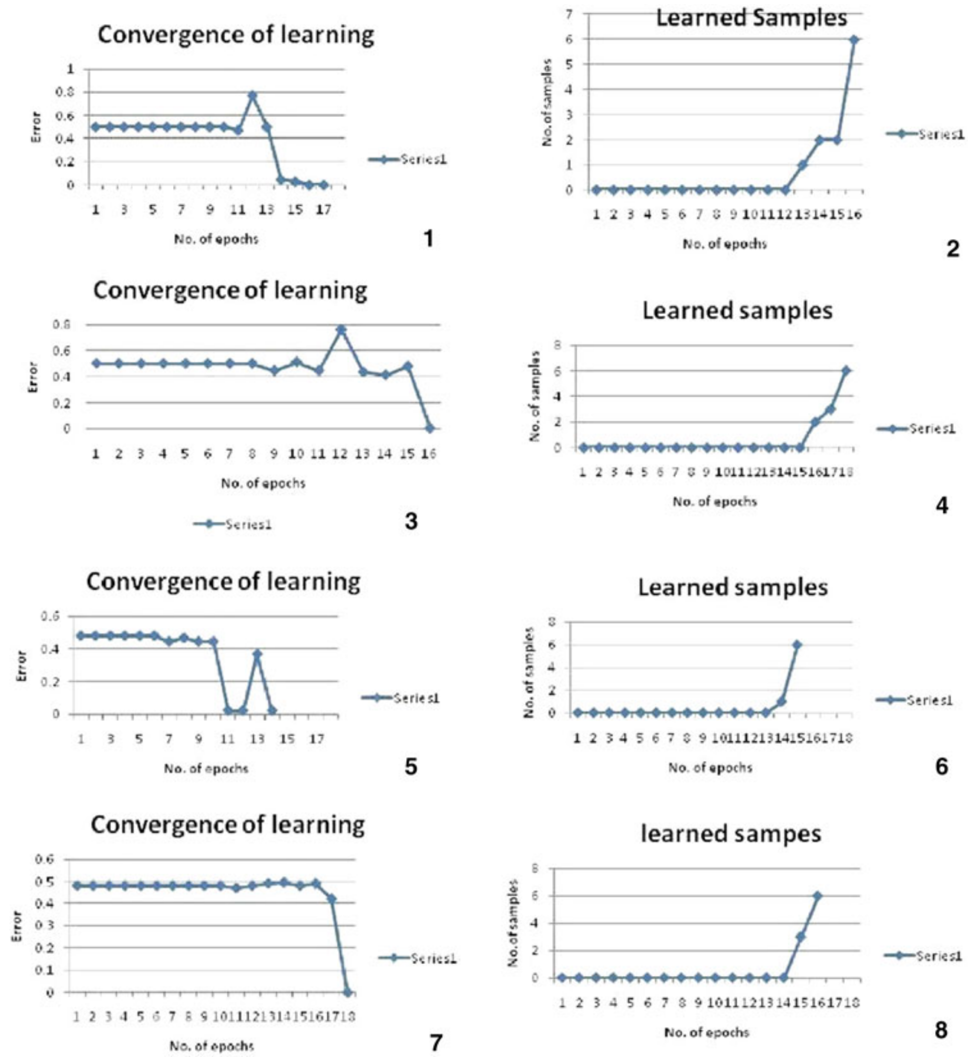


Fig. 15 The convergence of learning algorithm and Learn samples with respect to number of epochs

Table 4 Summary of the recognition results and the recognition rates

Gesture meaning	Number of test gesture	Successful recognition	Recognition rate (%)
Open	8	4	50
Close	9	7	77
Cut	9	8	88
Paste	10	6	60
Maximize	10	8	80
Minimize	10	7	70
Total	56	40	70.8

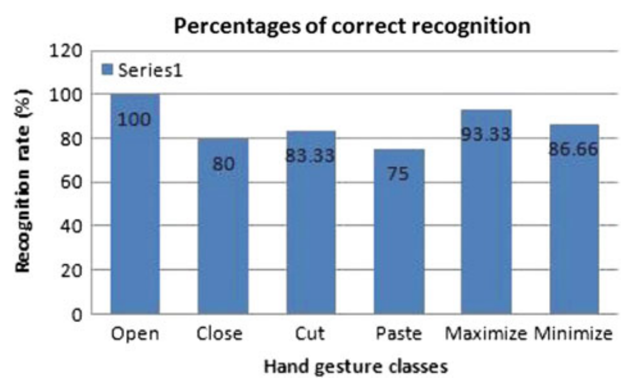


Fig. 16 Percentages of correct recognition

3.1.6 Artificial Neural Networks Training and Testing Phases

(a) Training phase

After the computation of feature vectors, each containing 10 translation, scaling, and rotation invariant elements characterized the complex moments for the hand gesture. Five similar neural network classifiers are trained with a data set containing 30 features vectors, and these vectors were computed from a set of 5 examples for each hand gesture performed by one subject.

Table 5 Complex moments values

Moment order	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
Open	45,660	0.11310	2,167.2	2,330.2	538.9	1,448.3	55.3	57.6	1,691.2	8,819
Close	44,652	0.02355	953.7	1,248.5	2,342.8	1,971.3	1,261.6	731.1	1,656.5	905.8
Cut	30,916	0.16550	7,553.2	1,300.5	2,793.3	1,545.0	3,230.4	1,249.5	4,125.3	1,247.2
Paste	37,186	0.14053	4,680.8	1,900.6	965.4	1,161.9	1,418.5	1,190.7	408.5	616.5
Maximize	28,965	0.17819	7,142.2	3,383.1	8,998.0	752.4	7,099.1	2,109.6	10,954.9	1,894.8
Minimize	43,866	0.08460	5,710.4	1,938.4	3,020.8	2,963.1	2,904.2	2,420.0	3,076.2	3,117.8

Table 6 Complex moments values after normalization

Moment order	M0	M1	M2	M3	M4	M5	M6	M7	M8	M9
Open	1	0.63	0.28	0.68	0.05	0.48	0.07	0.02	0.15	0.28
Close	0.97	0.13	0.12	0.36	0.26	0.66	0.17	0.30	0.15	0.29
Cut	0.67	0.92	1	0.38	0.31	0.52	0.45	0.51	0.37	0.40
Paste	0.81	0.78	0.61	0.56	0.10	0.39	0.19	0.49	0.03	0.19
Maximize	0.63	1	0.94	1	1	0.25	1	0.87	1	0.60
Minimize	0.96	0.47	0.75	0.57	0.33	1	0.40	1	0.28	1

Table 7 Parameters of back-propagation neural networks

Parameters	Values
Input layer	10 nodes
Output layer	6 nodes
No. of images for each training set	6
Stop error	0.01
Learning rate	0.9

The learning process for the back-propagation neural networks is accomplished using the following parameter for each one as shown in Table 7. The training results as shown in Fig. 17 show the convergence of learning algorithm for three back-propagation neural networks (first, fourth, and fifth) and the learned samples with respect to the number of epochs for each network.

(b) Testing Phase
After training the neural networks, performance is estimated by applying the testing set on the networks inputs and computing the classification error. The activation function used is binary-sigmoid which holds outputs always between 0 and 1. The testing process is as in the Sect. 3.1.3(b) of the previous method. In this phase, 84 hand gesture images are used to test the system. Each one of the six hand gestures has a number of samples in different light conditions and with effects of scaling, translation, and rotation. Each gesture has a table of recognition results and with neural network outputs for one gesture image (as an example) a summary of all the recognition results and the recognition rates for each of the six static hand gestures is presented in Table 8. These results are obtained using the formula (14) and the recognition rates with each class is shown in Fig. 18.

4 Conclusions

This paper addresses the problem of static hand gesture recognition. In this research, a computer vision algorithm is proposed that recognizes the six selected static hand gestures, namely, Open, Close, Cut, Paste, Maximize, Minimize, used for HCI. A supervised back-propagation multi-layer feed-forward neural network is used for recognition, with two types of features: hand contour and the hand complex moments. The recognition is made, however, without the need for using special hardware such as gloves.

It is difficult to find a highly accurate hand gesture recognition system that is capable of working under various conditions including varying illumination complex background and effects of scaling, translation and rotation by specific angles. However, this static hand gesture recognition algorithm for both methods, mentioned earlier, performs well in recognizing the six static hand gestures, with some instances of failure for both methods. In general, the conclusions resulted from the foregoing analyses are

Neural network with hand contour The results obtained applying this method are presented in Fig. 15. According to the results, this method takes a short time for convergence of learning algorithm of the neural networks. In Table 4, it can be observed that the method has sufficient accuracy with recognition rate of 70.83 % but is not invariant to translation

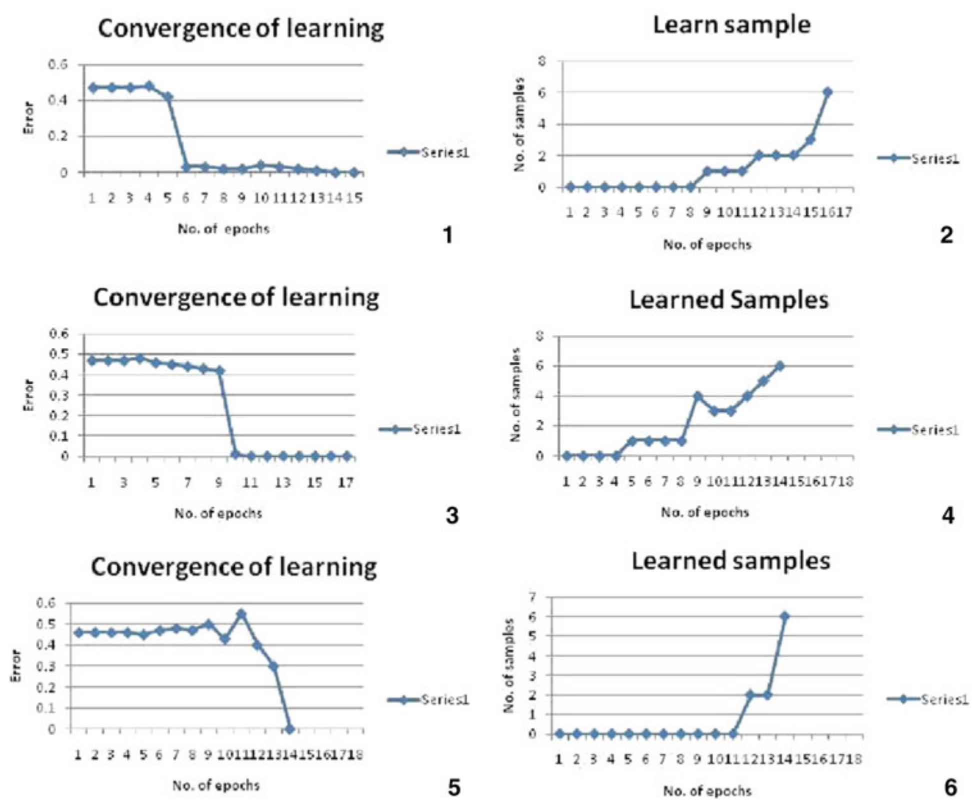


Fig. 17 The convergence of learning algorithm and Learn samples with respect to number of epochs

Table 8 Summary of the recognition results and the recognition rates

Gesture meaning	Number of test gesture	Successful recognition	Recognition rate (%)
Open	15	15	100
Close	15	12	80
Cut	12	10	83.3
Paste	12	9	75
Maximize	15	14	93.3
Minimize	15	13	86.6
Total	84	70	86.3

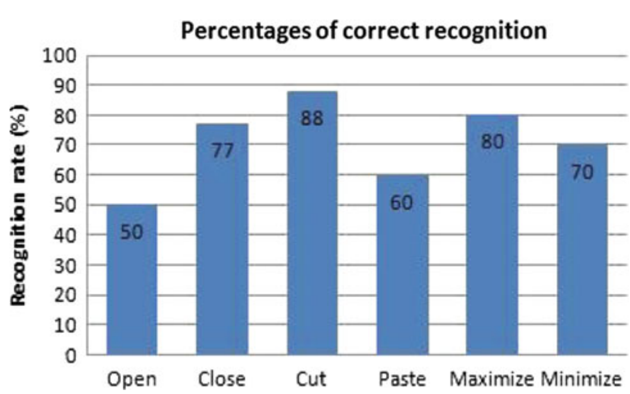


Fig. 18 Percentages of correct recognition

cases. Using pure hand contour is not enough for successful recognition results. It can be used to calculate feature vectors with parameters such as the area of hand shape and hand perimeters, and to find some relation between them.

Neural network with hand complex moments As displayed in Fig. 17, this method needs a long time for convergence of learning how to train the neural networks. The performance of this method, as shown in Table 8, with recognition rate of 86.38 % and with effects of scaling and translation and rotation by specific angles, is significantly better than the previous method for recognizing the six gestures. For both methods, however, the same gesture in different light conditions is not correctly recognizable because the distance between the camera and the hand gesture is not fixed.

References

1. Just, A.: Two-handed gestures for human–computer interaction. Ph.D. thesis (2006)
2. Symonidis, K.: Hand gesture recognition using neural networks. **68**, 5 (2000)
3. Manresa, C.; Varona, J.; Mas, R.; Perales, F.: Real-Time Hand Tracking and Gesture Recognition for Human–Computer Interaction. Computer Vision Centre, University Autonomic, Barcelona. (2000)
4. Wu, Y.; Huang, T.S.: Vision-based gesture recognition: a review. International Gesture Workshop, France (1999)

5. Dadgostar, F.; Barczak, A.L.C.; Sarrafzadeh, A.: A color hand gesture database for evaluating and improving algorithms on hand gesture and posture recognition. In: Research Letters in the Information and Mathematical Sciences, vol. 7, pp. 127–134. (2005)
6. Freeman, W.T.; Roth, M.: Orientation Histograms for hand gesture recognition. In: IEEE International Workshop on Automatic Face and Gesture Recognition, Zurich (1995)
7. Triesch, J.; von der Malsburg, C.: Robust classification of hand postures against complex backgrounds. In: Proceedings of Second International Conference on Automatic Face and Gesture Recognition. Killington (1996)
8. Naidoo, S.; Omlin, C.W.; Glaser, M.: Vision-Based Static Hand Gesture Recognition Using Support Vector Machines. University of Western Cape, Bellville (1998)
9. Licsár, A.; Szirányi, T.: Hand-gesture based film restoration. In: Second International Workshop on Pattern Recognition in Information Systems, pp. 95–103. Alicante, Spain (2002)
10. Hu, C.; Qinghu, M.; Liu, P.; Wang, X.: Visual gesture recognition for human–machine interface of robot teleoperation. In: Conference on Intelligent Robots and Systems, Las Vegas, Oct 2003. (2003)
11. Chang, C.C.; Chen, J.J.; Tai, W.; Han, C.C.: New approach for static gesture recognition. J. Inform. Sci. Eng. **22**, 1047–1057 (2006)
12. Parvini, F.; Shahabi, C.: An algorithmic approach for static and dynamic gesture recognition utilising mechanical and biomechanical characteristics. Int. J. Bioinforma. Res. Appl. **3**(1) (2007)
13. Vieriu, R.L.; Goras, B.; Goras, L.: On HMM static hand gesture recognition. In: 10th International Symposium on Signals, Circuits and Systems (ISSCS), pp. 1–4 (2011)
14. Swapna, B.; Pravin, F.; Rajiv, V.D.: Hand gesture recognition system for numbers using thresholding. In: Communications in Computer and Information Science, vol. 250, Part 3, pp. 782–786. (2011)
15. Kjrlfdrn, F.C.M.: Visual interpretation of hand gestures as a practical interface modality. Ph.D. thesis, Columbia University, New York (1997)
16. Winnemöller, H.: Practical gesture recognition for controlling virtual environments. Project for Bachelor of Science (Honours) of Rhodes University (1999)
17. Awcock, G.J.; Thomas, R.: Applied image processing. Macmillan Press LTD, London (1995)
18. Umbaugh, S.E.: Computer vision and image processing a practical approach using CVIP tools. Prentice Hall, Englewood Cliffs (1998)
19. Gonzalez, R.C.; Woods, R.E.: Digital Image Processing, 2nd edn. Wesley Publishing, Cambridge (2001)
20. Philips, D.: Image Processing. C, R and D Publications Inc., Lawrence (1994)
21. Pitas, I.: Digital Image Processing Algorithms and Application. Ph.D. thesis, Cornell University (1998)
22. Huang, J.: Spatial Image Indexing and Application. Ph.D. thesis, Cornell University (1998)
23. Joshi, G.; Sivaswamy, J.: A Simple Scheme for Contour Detection. Center for Visual Information Technology, IIIT, Hyderabad (2004)
24. Boussaid, K.; Beghdaddi, A.: A Contour Detection Method Based on Some Knowledge of the Visual System Mechanisms, Vision Interface. Rrois-Rivieres, Canada (1999)
25. Barczak, A.; Dadgostar, F.: Real-time hand tracking using a set of cooperative classifiers based on Haar-like features. Res. Lett. Inf. Math. Sci. **7**, 29–42 (2005)
26. AbdAlrazak, R.: Signature recognition using wavelet transform. Ph.D. Thesis, University of Technology (2004)
27. Abo-Mustafa, Y.; Psaltis, D.: Aspects of moment invariants. IEEE Trans. Pattern Anal. Mach. Intell. **6**(6), 698–706 (1984)
28. Musa, A.K.: Signature recognition and verification by using complex-moments characteristics. Masters thesis, University of Baghdad (1998)
29. Fausett, L.: Fundamentals of Neural Network'S Architectures, Algorithms and Applications. Prentice Hall Inc, Englewood Cliffs (1994)
30. Picton, P.: Neural Networks, 2nd edn. Palgrave, New York (2000)
31. Haykin, S.: Neural Networks a Comprehensive Foundation. Macmillan College Publishing Company, New York (1994)
32. Kinnebrock, W.: Neural Networks, Fundamentals, Applications, Examples, 2nd revised edn. Galgotia Publications, New Delhi (1995)