COMPUTATIONAL INTELLIGENCE TECHNIQUES IN VISUAL PATTERN RECOGNITION

By

PRAMOD KUMAR PISHARADY

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY
AT

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING,
NATIONAL UNIVERSITY OF SINGAPORE
4 ENGINEERING DRIVE 3, SINGAPORE 117576

MARCH 2012

Table of Contents

List of Tables

List of Figures

Abstract

Efficient feature selection and classification algorithms are necessary for the effective recognition of visual patterns. The initial part of this dissertation presents fast feature selection and classification algorithms for multiple feature data, with application to visual pattern recognition. A fuzzy-rough approach is utilized to develop a novel classifier which can classify vague and indiscernible data with good accuracy. The proposed algorithm translates each quantitative value of a feature into fuzzy sets of linguistic terms using membership functions. The fuzzy membership functions are formed using the feature cluster centers identified by the subtractive clustering technique. The lower and upper approximations of the fuzzy equivalence classes are obtained and the discriminative features in the dataset are identified. The classification is done through a voting process. Two algorithms are proposed for the feature selection, an unsupervised algorithm using fuzzy-rough approach and a supervised method using genetic algorithm. The algorithms are tested in different visual pattern classification tasks: hand posture recognition, face recognition, and general object recognition. In order to prove the generality of the classifier for other multiple feature patterns, the algorithm is also applied to cancer and tumor datasets. The proposed algorithms identified the relevant features and provided good classification accuracy, at a less

computational cost, with good *margin of classification*. On comparison, the proposed algorithms provided equivalent or better classification accuracy than that provided by a Support Vector Machines classifier, at a lesser computational time.

The later part of the thesis presents the results of the utilization of computational model of visual cortex for addressing problems in hand posture recognition. The image features have invariance with respect to hand posture appearance and its size, and the recognition algorithm provides person independent performance. The features are extracted in such a way that it provides maximum inter class discrimination. The real-time implementation of the algorithm is done for the interaction between the human and a virtual character *Handy*.

A system for the recognition of hand postures against complex natural backgrounds is presented in the last part of the dissertation. A Bayesian model of visual attention is utilized to generate a *saliency map*, and to detect and identify the hand region. Feature based visual attention is implemented using a combination of high level (shape, texture) and low level (color) image features. The shape and texture features are extracted from a skin color map, using the computational model of the visual cortex. The skin color map, which represents the similarity of each pixel to the human skin color in HSI color space, enhanced the edges and shapes within the skin colored regions. The hand postures are classified using the shape and texture features, with a support vector machines classifier. The algorithm is tested using a newly developed complex background hand posture dataset namely NUS hand posture dataset-II. The experimental results show that the algorithm has a person independent performance, and is reliable against variations in hand sizes. The proposed algorithm

provided good recognition accuracy despite clutter and other distracting objects in the background, including the skin colored objects.

Acknowledgements

With immense pleasure I express my gratitude and indebtedness to my supervisors Assoc Prof. Prahlad Vadakkepat and Assoc Prof. Loh Ai Poh for their excellent guidance, invaluable suggestions, and the encouragement given at all the stages of my doctoral research. In particular, I would like to thank them for sharing their scientific thinking and shaping my own critical judgment capabilities, which helped me to sift out golden principles from the dross of competing ideas. The freedom given by them for independent thinking by imparting confidence on me helped my growth as an independent and skilled researcher.

Many thanks goes to the other members of my thesis panel, Assoc Prof. Abdullah Al Mamun, and Assoc Prof. Tan Woei Wan for their patience, timely guidance, and advices. I would like to express sincere appreciation to my senior Dr. Dip Goswami for the many constructive and insightful discussions and for his friendship. My gratitude also goes to Ms. Quek Shu Hui Stephanie, and Ms. Ma Zin Thu Shein for helping me to implement the algorithm in real-time, and to develop a new hand posture database for experimental analysis. Also I would like to thank my

room mate Padmanabha Venkatagiri for his help in analyzing the computational complexity of the proposed algorithms. Then there are the colleagues with whom I have had the pleasure of exchanging technical ideas as well as witty repartee: Mr. Ng Buck Sin, Mr. Hong Choo Yang Daniel, Mr. Yong See Wei, Mr. Christopher, and Mr. Jim Tan. Finally, I show my appreciation to Dr. Tang and the lab officer Mr. Tan Chee Siong for their support and friendly behavior.

I express my deepest appreciation to all the members of the department of Electrical and Computer Engineering, for the wonderful research environment and immense support provided during my doctoral studies.

Lastly I thank the chief supporters of my doctoral studies: my beloved wife, parents, and other family members for their encouragement, understanding and support in every aspects of life.

Chapter 1

Introduction

Recognition of visual patterns has wide applications in surveillance, interactive systems, video gaming, and virtual reality. The unresolved challenges in visual pattern recognition techniques assure wide scope for research. Image feature extraction, feature selection, and classification are the different stages in a visual pattern recognition task. The efficiency of the overall algorithm depends on the individual efficiencies of these stages.

Hand gestures are one of the most common body language used for communication and interaction among human beings. Because of the naturalness of interaction, hand gestures are widely used in human robot interaction, human computer interaction, sign language recognition, and virtual reality. The release of the motion sensing device Kinect by Microsoft demonstrates the utility of tracking and recognition of human gestures in entertainment. Visual interaction using hand gestures is an easy and effective way of interaction, which does not require any physical contact and does not get affected by noisy environments. However complex scenery and cluttered backgrounds make the recognition of hand gestures

difficult.

Recognition of visual patterns for real world applications is a complex process that involves many issues. Varying and complex backgrounds, bad lighted environments, person independent recognition, and the computational costs are some of the issues in this process. The challenge of solving this problem reliably and efficiently in realistic settings is what makes research in this area difficult.

1.1 Overview

A typical image pattern recognition pipeline is shown in Fig. 1.1. Image feature extraction, feature selection, and classification, which are the main stages in a visual pattern recognition task, are the focus of this thesis. Novel algorithms are proposed for feature extraction, feature selection, and classification using computational intelligence techniques.

The main goal of the research reported in this dissertation is to propose computationally efficient and accurate pattern recognition algorithms for Human-Computer Interaction (HCI). The main area of focus is hand posture recognition. However the research conducted has several directions. The thesis proposes two feature selection and classification algorithms based on fuzzy-rough sets, and neuro-biologically inspired hand posture recognition algorithms.

Fuzzy and rough sets are two computational intelligence tools used for

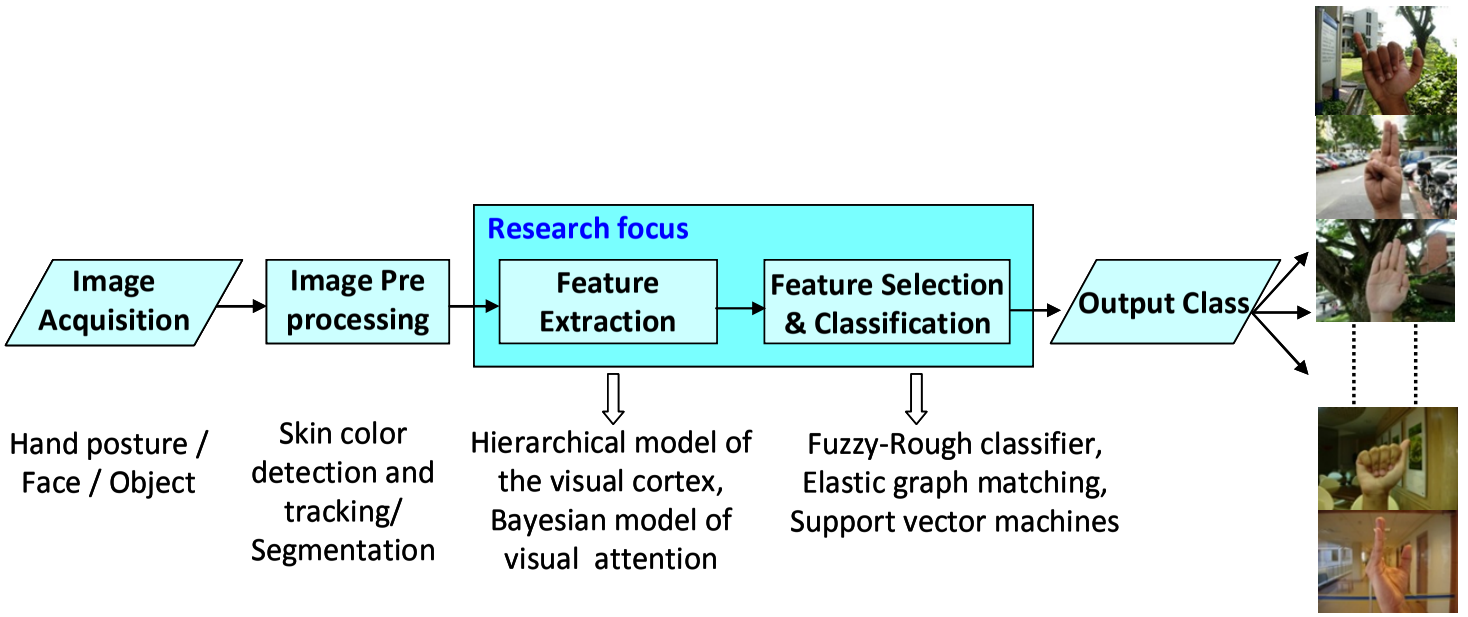


Figure 1.1: Visual pattern recognition pipeline.

making decision in uncertain situations. This work utilizes the fuzzy-rough approach to propose novel feature selection and classification algorithms for datasets with large number of features. The presence of large number of features makes the classification of multiple feature datasets difficult. The proposed algorithms are simple and effective in such classification problems. The feature selection and classification algorithms proposed in the thesis are applied to different visual pattern recognition tasks: hand posture, face, and object recognition. In order to prove the generality of the classifier, the algorithms are also applied to cancer and tumor classification problems. The proposed classifier is effective in cancer and tumor classification, which is useful in the biomedical field.

The visual processing and pattern recognition capabilities of the primate brain is yet to be understood well. The human visual system rapidly and effortlessly recognizes a large number of diverse objects in cluttered, natural scenes and identifies specific patterns, which inspired the development of computational models of biological vision systems. These models can be utilized for addressing problems in conventional pattern recognition. This thesis utilizes a computational model of the ventral stream of visual cortex for the recognition of hand postures. The features extracted using the model have invariance with respect to hand posture appearance and size, and the recognition algorithm provides person independent performance. The image features are extracted in such a way that it provides maximum inter class discrimination.

The thesis addresses the complex natural background problem in hand posture recognition using a Bayesian model of visual attention. A saliency

map is generated using a combination of high and low level image features. The feature based visual attention helps to detect and identify the hand region in the images. The shape and texture features are extracted from a skin color map, using the computational model of the ventral stream of visual cortex. The color features used are the discretized chrominance components in HSI, YCbCr color spaces, and the similarity to skin color map. The hand postures are classified using the shape and texture features, with a support vector machines classifier.

1.2 Problem Statement

Hand postures are widely used for communication and interaction among human. The same hand posture shown by different persons varies as the human hand is highly articulated and deformable, and has varying sizes. Other factors which affect the appearance of the hand postures are the view point, scale, illumination and the background. Human visual system has the capability to recognize visual patterns despite these variations and noises. The real world application of computer vision based hand posture recognition systems necessitates an algorithm which is capable of handling the variations in hand posture appearance and the distracting patterns. At the same time, the algorithm should be capable to distinguish different hand posture classes which look similar. The biologically inspired object recognition models provide a trade-off between the selectivity and invariance. The current work utilizes a computational model of the visual cortex for extracting the image features which contains the pattern to be recognized.

The features extracted using the computational model provides good recognition accuracy. However the model (the feature extraction process) has high computational complexity. A major limitation of the model in real-world applications is its processing speed [85].

The visual features at the output of the feature extraction stage are large in number. In general classification of multiple feature datasets is a difficult process. In addition, the features extracted from images of different classes that looks similar have vague and indiscernible classification boundary. These issues lead to the need for an efficient feature selection algorithm, and a computationally simple classifier that can classify vague and indiscernible data with good accuracy.

The poor performance against complex natural backgrounds is another major problem in hand posture recognition. Skin color based segmentation improves the performance to a certain extent. However the conventional skin color based algorithms fail when the complex background contains skin colored regions.

1.3 Major Contributions

The major contribution of the dissertation is a computationally efficient and accurate feature selection and classification algorithm for multiple feature datasets. The concept of fuzzy-rough sets is utilized to develop a simple and effective classifier that can classify vague and indiscernible data with good accuracy. The proposed algorithm has a polynomial time complexity. The feature selection algorithm identified the discriminative

features in the dataset, which enhanced the shape selectivity and reduced the computational burden of the pattern recognition algorithm. The feature selection and classification algorithms are applied to hand posture, face, and object recognition.

Two hand posture recognition algorithms are proposed utilizing a standard model of the visual cortex for the feature extraction. The algorithms are robust against variations in hand posture appearance and size, and provided person independent performance. The features are extracted in such a way that it provides good interclass discrimination even between the classes which look similar. The proposed algorithm improved the processing speed by identifying and selecting the relevant and predictive features of the image. The selection of relevant features improved both feature extraction and classification time, which makes the algorithm suitable for real-time applications.

Another major contribution of the dissertation is an algorithm for hand posture recognition against complex natural backgrounds. A Bayesian model of visual attention is used for focussing the attention on the hand region and to segment it. Feature based attention is implemented utilizing a combination of color, texture, and shape based image features. The proposed algorithm improved the recognition accuracy in the presence of clutter and other distracting objects, including skin colored objects.

Two new hand posture datasets, the NUS hand posture dataset I & II (10 class simple background and 10 class complex background respectively, with variations in hand sizes and appearances), are developed for

the experimental evaluation of the proposed hand posture recognition algorithms.

The feature selection and classification algorithm proposed in the thesis is successfully applied for the predictive gene identification and classification of cancer and tumor (which are non-visual patterns). This shows the utility of the algorithm in multi feature classification problems in biomedical field.

1.4 Organization

The rest of the thesis is organized as follows. Chapter 2 provides survey of the literature in the hand posture recognition and fuzzy-rough classification fields, and a brief explanation of the biologically inspired feature extraction system. Chapter 3 describes a fuzzy-rough discriminative feature selection and classification algorithm, with applications to image pattern (object, hand posture, face) classification and cancer classification. Fuzzy-rough sets based hand posture and face recognition algorithm is proposed in Chapter 4. Chapter 5 explains the neuro-biologically inspired approaches for hand posture recognition. The problem of complex backgrounds in hand posture recognition is addressed in Chapter 6 (Chapters 3 and 4 focus on the feature selection and classification aspects, whereas Chapters 5 and 6 focus on the feature extraction aspect). The final chapter concludes the thesis with a summary of the work done, and a statement of possible future research directions.

Chapter 2

Literature Survey

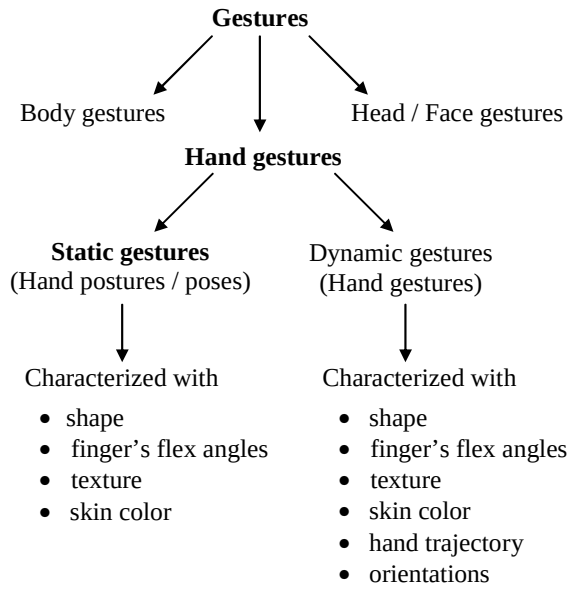
This chapter provides a literature survey on the tools and techniques in the background of this thesis. A detailed review of hand gesture recognition techniques, a brief survey on fuzzy-rough classifiers, and a brief study of the biologically inspired feature extraction system are presented.

2.1 Hand Gesture Recognition

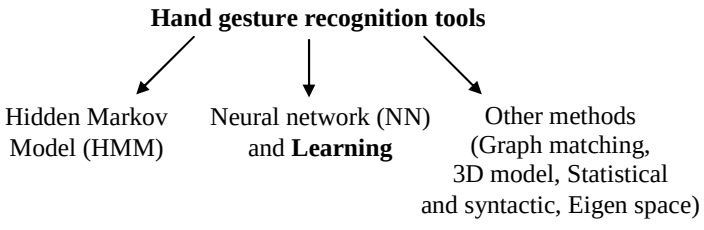
This section focuses on the developments in the hand gesture recognition and classification field during the last decade. A categorized analysis of different hand gesture recognition tools and a list of the available hand gesture databases are provided.

Gestures are expressive, meaningful body motions involving physical movements of the fingers, hands, arms, head, face, or body [57]. Gestures are classified based on the moving body part (Fig. 2.1(a)). There are two types of hand gestures; *static* and *dynamic* gestures. Static hand gestures (hand postures / poses) are those in which the hand position does

not change during the gesturing period. Static gestures mainly rely on the shape and the flexure angles of the fingers. In dynamic hand gestures (hand gestures), the hand position is temporal and it changes continuously with respect to time. Dynamic gestures rely on the hand trajectories and orientations, in addition to the shape and fingers flex angles. Dynamic gestures, which are actions composed of a sequence of static gestures, can be expressed as a hierarchical combination of static gestures.



(a)



(b)

Figure 2.1: Classification of (a) gestures and (b) hand gesture recognition tools. The proposed algorithm recognizes static hand gestures, using a learning based approach.

There exist several reviews on hand modeling, pose estimation, and

gesture recognition [20, 57, 61, 63, 108, 121]. [57] provided a survey of different gesture recognition methods, which considered the hand and arm gestures, the head and face gestures, and, the body gestures. Hand modeling and three dimensional (3-D) motion based pose estimation methods are reviewed in [20]. An analysis of sign languages, grammatical processes in sign gestures, and issues relevant to the automatic recognition of sign languages are discussed in [61]. The classification schemes in glove based and vision based sign language recognition are also discussed in their survey. Another survey on hand gesture recognition techniques is provided in [63], which discusses gesture modeling, interpretation, and recognition. The developments till the year 1997 was considered in their review whereas [108] give another review of vision based gesture recognition which covered developments till the year 1999. An elaborate and categorized analysis of hand gesture recognition techniques is done in the present study which makes this survey unique. The hand gesture recognition methods are classified and analyzed according to the tools used for recognition. A list of available hand gesture databases and a comparison of different hand gesture recognition methods are also provided.

2.1.1 Different Techniques

There are different methods for hand gesture recognition. The initial attempts in hand gesture recognition utilized mechanical devices that directly measure hand and / or arm joint angles and spatial position, using glove-based devices. Later vision based non-contact methods developed. Vision-based hand gesture recognition techniques can be broadly divided

into two categories, appearance-based approaches and 3-D hand model-based approaches. Appearance-based approaches utilize features of training images to model the visual appearance of the hand, and compare these parameters with the extracted features of testing images. Three-dimensional hand model-based approaches rely on a 3-D kinematic hand model, by estimating the angular and linear parameters of the kinematic model.

The tools used for vision based hand gesture recognition can be classified into three categories (Fig. 2.1(b)). They are *1)* Hidden Markov Model (HMM) based methods [9,41,50,79,113,117], *2)* Neural network (NN) and learning based methods [3,19,22,29,52,74,76,77,91,92,109,114,115,119], and *3)* Other methods (Graph algorithm based methods [75, 96–98], 3D model based methods [4, 49, 103, 116], Statistical and syntactic methods [10, 105], and Eigen space based methods [14, 62]).

Hidden Markov Model based Methods

Hidden Markov Model (HMM) is the most widely used hand gesture recognition technique [9,41,46,50,53,79,113,117]. It is a useful tool for modeling the spatiotemporal variability of gestures in a natural way [30]. HMM is a statistical model in which the system being modeled is assumed to be a Markov process with unknown parameters. A Markov process is a mathematical model of a system in which the likelihood of a given future state, at any given moment, depends only on its present state, and not on any past states. A HMM is employed to represent the statistical behavior of an observable symbol sequence in terms of a network of states [9]. For each observable symbol, it can be modeled as one of the states of the

HMM, and then the HMM either stays in the same state or moves to another state based on a set of state transition probability associated with the state. The observable event is a probabilistic function of the hidden states, and so the hidden parameters in the HMM are identified using the observable data, and these parameters are used for pattern recognition.

HMM based dynamic hand gesture recognition is done in [9] using the spatial and temporal features of the input image. Fourier descriptor and optical flow based motion analysis are used to characterize spatial and temporal features respectively. The work also proposes a real time hand gesture tracking technique which can track the moving hand and then extract the hand shape from complex backgrounds. The algorithm is tested using 20 different hand gestures selected from the Taiwanese Sign Language(TSL) and more than 90% average recognition accuracy is achieved.

Normal HMM based recognizer identifies the best likelihood gesture model for a given pattern. However the similarity of the pattern to the reference gesture cannot be guaranteed unless the likelihood value is high enough. Lee *et al.* [50] addressed this problem by introducing the concept of *threshold model* using HMM, to filter out non-gesture patterns, among dynamic hand gestures. A gesture is described as a spatio-temporal sequence of feature vectors that consist of the direction of hand movement. The threshold model approves or rejects the pattern as a gesture and a gesture is recognized only if the likelihood of the best gesture model is higher than that of the threshold model. The method detects reliable end point of a gesture and finds the start point by backtracking. However the number of states in the threshold model is large which increases the computational cost and slows down the recognition speed. The authors

alleviated this problem by reducing the number of states of the threshold model using the relative entropy, which is often used as a measure of the distance between two probability distributions. Pairs of states with the least distance are merged to reduce the computational requirements. Ten dynamic hand gestures, which corresponds to 10 most frequently used browsing commands in Power Point presentation, are considered in their work and the method extracted trained gestures from continuous hand motion with 93.14% reliability. Similar to [50], [46] uses a feature vector created from the direction of hand movement to model the dynamic hand gestures using HMM. The system designed is used for controlling a mobile robot.

Marcel *et al.* [56] proposed to use an extension of HMM, viz. the Input / Output Hidden Markov Model (IOHMM), for hand gesture recognition. An IOHMM is based on a non-homogeneous Markov chain where emission and transition probabilities depend on the input. In contrast the HMM is based on homogeneous Markov chains since the dynamics of the system are determined only by the transition probabilities, which are time independent. Compared to HMMs, IOHMM is a discriminative approach as it directly models posterior probabilities. However [41] compared HMM and IOHMM, and concluded that HMM have better performance than IOHMM. They performed the experiments on larger databases, ranging from 7 to 16 dynamic gesture classes, whereas [56] considered only 2 classes. [41] also contributed two hand gesture databases [40], one containing both one and two handed gestures, and the second containing only two handed gestures.

An implementation of HMM, for dynamic gesture recognition, using

the combined features of hand location, angle and velocity is provided in [117]. The hand localization is done by skin-color analysis and the hand tracking is done by finding and connecting the centroid of the moving hand regions. The extracted features are quantized so as to obtain discrete symbols to input to the HMM. From the gesture trajectory the discrete symbols are made using the k-means vector quantization algorithm. The k-means clustering algorithm is adopted to classify the gesture tokens into different clusters in the feature space. Experiments are done using 48 classes of gestures for 36 alphanumeric characters and 12 graphic elements. A set of 2400 trained gestures and 2400 untrained gestures are used for training and testing of the algorithm respectively. Accuracy of 98.96% and 93.25% are achieved for training and testing datasets respectively, while the features are combined in cartesian system. The work concludes that angle feature is the most effective one in providing better accuracy, among the three features, location, angle, and velocity. It also provided an analysis of the variation of accuracy, with respect to the number of feature codes, and identified the best number of feature codes. A similar HMM implementation, which utilizes angles of motion along the trajectory of the centroid of hand is provided in [53]. The algorithm recognized 26 alphabet (A-Z) hand gestures with an average recognition rate of 90%.

A robust system for the gesture based control of a robot is developed using a combination of HMM based temporal characterization scheme, static shape recognition, and Kalman filter based hand tracking in [79]. The system uses skin color for static shape recognition and tracking. The static shape recognition is performed using contour discriminant analysis.

Table 2.1: Hidden markov model based methods for hand gesture recognition: A comparison

Work	**S/D**	**Features used**	**Classification method**	**# classes**	**Accuracy (%)**	**Application**
Chen *et al.* [9]	D	fourier descriptors/ optical flow	HMM	20	above 90	Taiwanese sign language
Lee *et al.* [50]	D	direction of hand movement	HMM	10	93.14	browsing commands in power point presentation
Just *et al.* [41]	D	3D trajectory, hand displacement, color & shape of hand blob	HMM& IOHMM	16&7 (2 datasets)	75&63, 98&43	Interact-play, manipulation
Yoon *et al.* [117]	D	hand location, angle & velocity	HMM	48	93.25	HCI- recognizing alphanumeric characters & graphic elements
Ramamoo -rthy *et al.* [79]	D	hand shape & hand motion	HMM	5	79	remote robot control

S-static, **D**-dynamic, **HMM**-Hidden markov model, **IOHMM**-Input / Output hidden markov model, **HCI**-Human computer interaction

A Kalman filter based estimator is utilized in the hand contour tracker. The tracker provides temporal characteristics of the gesture and the output of the tracker is used for classifying the nature of the motion. Shape classification information is provided by the contour discriminant-based classifier. These symbolic descriptors, corresponding to each of the gesture, are utilized for training the HMM. The system can reliably recognize dynamic gestures in spite of motion and discrete changes in hand poses. It also has the ability to detect the starting and ending of gesture sequences in an automated fashion. The original contributions of the work are, a novel technique for combining shape-motion parameters and system level techniques, and optimizations for the achievement of real-time gesture recognition. The system is tested using five dynamic gestures, which is associated with five different functions needed for robot motion. This method explicitly utilizes hand shape as a feature for gesture identification. The use of hand shape makes it easier for the gesturer to remember the commands, which increases the user friendliness of the system.

[113] proposes an important and complex application of HMM in Human-Robot Interaction (HRI) viz. whole-body gesture recognition. A set of features, encoding the angular relationship between a dozen body parts in 3-D is used to describe the gesture and this feature vector is then used in the HMM. A model reduction is done using the relative entropy, similar to that done in [50]. The whole-body gesture recognition is outside the scope of the present study.

Neural Network and Learning Based Methods

Zhao *et al.* [119] proposed recursive induction learning based on extended variable-valued logic in hand gesture recognition. Inductive learning is a powerful approach to knowledge acquisition by inducing rules from sets of examples or sets of feature vectors. The paper modified and extended the old concept of Variable-Valued Logic into Extended Variable-valued Logic (EVL) which provides a more powerful representation capability. The Star concept is also extended into a more general concept: R-Star. Based on EVL and R-Star, a heuristic algorithm viz. RIEVL (Rule Induction by Extended Variable-valued Logic) is developed. RIEVL can learn rules not only from examples but also from rule sets and can produce more compact rules than other induction algorithms. The ability of RIEVL to abstract reduced rule sets is critical to efficient gesture recognition. This capability allows to apply a large feature set to hand poses representing a particular gesture during training-time, and to derive a reduced rule set involving a subset of the training-time feature set to be applied at recognition-time. The algorithm is capable of automatically determining the most effective features. The system also determines a subset of features that are salient to the recognition task, which reduces the number of features that need to be computed at recognition-time. This is critical in realtime vision systems because the rule size and number of features computed directly impact the performance. RIEVL is well suited for gesture pose recognition because recursive learning allows refining the gesture coding for individuals and variable-valued logic permits multi-valued feature representation of gesture poses. The algorithm is tested using 15 static hand gestures. It provided 100% recognition accuracy for training

images and up to 94.4% accuracy for test set, outperforming all other inductive algorithms. This showed the efficacy of the system on real image data which has variations in the hand pose.

A time delay neural network is used in [114,115] to learn the 2D motion trajectories, for the classification of dynamic hand gestures. 2D motion trajectories are extracted by computing pixel matches between consecutive image pairs, after finding the affine transformations between consecutive frames. A multi-scale segmentation is performed to generate homogeneous regions in each frame. Such region based motion algorithms perform well in situations where intensity-based methods fail. For example, motion information in areas with little intensity variation is contained in the contours of the regions associated with such areas. The motion segmentation algorithm computes correspondences for such regions and finds the best affine transformation that accounts for the change in contour shape. The affine transform parameters for region at different scales are used to derive a single motion field, which is then segmented to identify differently moving regions between two frames. The 2D motion trajectories are then learned using a time-delay neural network (TDNN). TDNN is a multilayer feedforward network that uses shift windows between all layers to represent temporal relationships between events in time. An input vector is organized as a temporal sequence and at any instance only the portion of an input sequence within a time window is fed to the network. TDNN is a dynamic classification approach in that the network sees only a small window of the input motion pattern and this window slides over the input data while the network makes a series of local decisions. These local decisions have to be integrated into a global decision at

a later time. There are two good properties about TDNN. First, TDNN is able to recognize patterns from poorly aligned training examples. Second, the total number of weights in the network is relatively small since only a small window of the input pattern is fed to TDNN at any instance. In other words, TDNN has small receptive fields. This in turns helps reduce training time due to a small number of weights in each receptive field. The algorithm is tested using 40 hand gestures of American Sign Language. Best accuracy achieved for training set is 98.14% and that for test set is 99.02%.

[91] proposed a neuro-fuzzy algorithm for spatio-temporal hand gesture recognition. They used sensor gloves for sensing the hand position (not a vision based method). However the recognition algorithm can be utilized in vision based gesture recognition, with the appropriate visual features of the image. The approach employs a powerful method based on hyper rectangular composite neural networks (HRCNNs) for selecting templates. Templates for each hand shape are represented in the form of fuzzy IF-THEN rules that are extracted from the values of synaptic weights of the corresponding trained HRCNNs.

A novel approach to user independent static hand gesture recognition system is proposed in [51, 52]. The system is made adaptive to the user by on-line supervised training. Any non-trainer users will be able to use the system instantly, and, if the recognition accuracy decreases only the faulty detected gestures be retrained realizing fast adaptation. A supervised training method corrects the unrecognized gesture classes and an unsupervised method continuously runs to follow the slight changes in

gesture styles. These training methods are embedded into the recognition phase and the reference classes can be modified during the system operation. There is no need to retrain all the gestures of the vocabulary and the training rules are simple. The system is implemented as a camera-projector system in which users can directly interact with the projected image by hand gestures, realizing an augmented reality tool in a multi-user environment. The emphasis is given on the novel approach of dynamic and quick follow-up training capabilities instead of handling large pre-trained databases. During experiments for the recognition of 9 static hand gestures, only the initial user trained all the gestures and the subsequent users corrected the recognition accuracy through interactive training when any of the gesture classes had low recognition rates. From experimental results it is seen that when the trainer and tester are the same person, the recognition rates are above 99%. If the trainer and tester users are different, the recognition rate varied from 87 to 99%. However the interactive training improved the recognition rate (more than 98%).

[59] proposed a combination of hidden Markov model (HMM) and recurrent neural networks (RNN) for better classification accuracy than that achieved using either HMM or RNN. A comparison of HMM and RNN based methods is provided in the paper. The features used are based on fourier descriptors and both static and dynamic gestures are considered. The system is configured to interpret user's gestures in real-time, to manipulate windows and objects within a graphical user interface, using 14 hand gestures. The processing is done in two stages. In the first stage, a radial-basis function (RBF) network is used to obtain a likelihood of

the basic hand pose. Gesture recognition is implemented in the second stage using two independent classifiers, HMM and RNN. Such a modular approach leads to a robust gesture recognition method. At the same time, the computation time is reduced both in the training and recognition stages. Input to the classifiers in the second stage are the hand pose likelihood vectors from each frame (output of first stage) and the motion vector of the hand between the current and previous frames. Outputs from the two classifiers are combined by using a linear output layer to give final gesture recognition result. The first stage (pose classifier) achieved a classification accuracy of 90.9%. The second stage (gesture recognizer) achieved recognition accuracies 90.2% and 89.8% for HMM and RNN based recognizers respectively. The combination of HMM and RNN classifier gave an improved accuracy of 91.9%.

An unsupervised learning algorithm viz. distributed locally linear embedding (DLLE) is proposed in [22], for static gesture recognition and dynamic gesture tracking, by extracting the intrinsic structure of data, such as neighborhood relationship. A probabilistic neural network (PNN) is employed for static gesture classification. Locally linearly embedding (LLE) is an unsupervised learning algorithm that attempts to map high-dimensional data to low-dimensional space while preserving the neighborhood relationship. The paper modified LLE to DLLE, to discover the inherent properties of the input data, by noticing that some relevant pieces of information are distributed. The input data are mapped to low dimensional space where both the local neighborhood relationship and some global distributions are preserved. According to the DLLE algorithm, the distances between the projected data points in low-dimensional

Table 2.2: Neural network and learning based methods for hand gesture recognition: A comparison

Work	**S/D**	**Features used**	**Classification method**	**# classes**	**Accuracy (%)**	**Application**
Zhao *et al.* [119]	S	Multi-valued features	RIL	20	94.4	gesture commands
Yang *et al.* [115]	D	2D motion field / trajectory	NN	40	99.02	American sign language
Licsar *et al.* [52]	S	Discrete fourier transform based distance matric	nearest neighbour / maximum likelihood	9	above 98	gesture commands
Chan *et al.* [59]	S & D	fourier descriptors (represents shape of hand blob)	RBF, HMM & RNN	14	91.9	manipulation of objects in windows/ graphical user interface
Ge *et al.* [22]	S	geometric distance	DLLE / PNN	14	93.2	same dataset as in [59]
Teng *et al.* [92]	S	intrinsic geometry of hand	SLLE	30	90	Chinese sign language

S-static, **D**-dynamic, **RIL**-recursive induction learning, **NN**-neural network, **RBF**-radial basis function, **RNN**-recurrent neural networks, **PNN**-probabilistic neural network, **DLLE**-distributed locally linear embedding, **SLLE**-supervised locally linear embedding

space depend on the similarity of the input images. The images which are similar are projected with a small distance while the images that differ greatly are projected with a large distance. Based on the distances in low dimensional space, probabilistic neural network (PNN) is used to classify different static gesture images. A skeleton model of the hand is used for recognition and tracking. PNN is based on Bayes theorem, having higher training speed, good classification accuracy, and negligible retraining time. Maximum accuracy of 93.2% is achieved for classification of 14 static gesture classes, which outperformed HMM and NN based methods in [59].

Supervised learning in LLE algorithm is introduced in [92], for recognizing static gestures in Chinese sign language (CSL). LLE is used for the feature extraction process and and it is suggested that the method is suitable for real-time applications. Supervised LLE (SLLE) makes use of the class label information during training. Hand is detected using skin color and intrinsic geometry of the hand is used for gesture recognition. The algorithm achieved 90% average recognition rate and has some robustness against different lighting conditions and backgrounds.

Graph Algorithm Based Methods

Starting from the late seventies, Graph-based techniques are used as as a powerful tool for pattern representation and classification. After the initial enthusiasm, graph algorithms have been practically left unused for a long period of time. This is probably due to the high computational costs of graph algorithms, which still remains an unresolved problem. However, the use of graphs in computer vision and pattern recognition

is obtaining a growing attention from the research community, recently. This is because the computational cost of the graph-based algorithms is now becoming compatible with the computational power of new generation computers [13].

Elastic graph matching (EGM), a type of graph matching, is a neurally inspired pattern recognition architecture [47]. EGM has the inherent ability to handle geometric distortions, does not require a perfectly segmented input image, and can elegantly represent the variances in object appearance [97]. These advantages makes EGM a powerful tool for gesture recognition applications. However there are unresolved challenges like flexibility in the matching process and realtime performance, which necessitates further research. A survey of graph algorithms used for hand gesture recognition, the present trends, and the scope for further research is relevant in this context.

In a typical graph representation, regions of the image are represented by *vertices* in the graph. These vertices are related to each other by *edges*, which express structural relationships between objects. Vertices and edges are usually attributed. Triesch *et al.* employed [94, 96–98] the elastic graph matching (EGM) technique to develop person independent static hand gesture recognition system against complex backgrounds. Hand postures are represented by labeled graphs with an underlying two- dimensional topology. Attached to the nodes are *jets*, which is a local image description (image feature) based on Gabor filters. This approach achieved scale-invariant and user-independent recognition, without the need for hand segmentation. Different hand postures are represented as attributed graphs and comparisons are made between model graphs (in

the database) and data graph (corresponding to the real-time image). The nodes are compared using a similarity function, and the pattern is recognized by calculating the average node similarities.

Bunch graphs [107] are used to model the variability in object appearance. The natural variability in the attributes of corresponding points in several images of the same object or a class of objects is captured, by labeling each node of a graph with a set or *bunch* of attribute values, extracted from corresponding points in different sample images. This method is also used to model complex backgrounds of the image [98]. For the matching process, each of the attribute value in the bunch is compared with the local image information in the data graph, and the maximum of the similarities is taken as the similarity of the bunch graph. The algorithm also proposes the use of multiple features for gesture recognition. The EGM is done using three features, 1) conventional Gabor jets, 2) color average and 3) color Gabor jets (convolutions of Gabor filter with images expressing each pixel's similarity to skin color). In order to find similarity between two nodes, similarities of the corresponding features of nodes are considered. These are then combined by computing a weighted average to find the net node similarity. The algorithm is tested using 12 hand postures, performed by 19 persons, against simple and complex backgrounds. Accuracies of 92.9% and 85.8% are achieved for simple and complex background images respectively.

Topology / 3D model Based Methods

Three dimensional model fitting is used in [103] for hand pose estimation that can be used for vision based human interfaces. However the

algorithm requires faster implementation for real-time processing. The method estimates all joint angles and hand pose is reconstructed as a voxel model. Then model fitting is done between the hand model and the voxel model, in the 3-D space. The method uses only geometric information of hand model and the voxel model for model fitting and do not need any heuristic or priori information. The feasibility of the method is tested using simulated and real hand poses. However the accuracy and processing speed of the algorithm can be improved. This method utilized kinematic model of the hand for gesture recognition, whereas a computer vision model is introduced in [116]. It avoids the complexity in estimation of the angular and linear parameters of the kinematic model. [116] suggests topological features of the hand for 3D hand gesture recognition. The hand is segmented from complex background using a restricted coulomb energy (RCE) neural network based on color segmentation method. The edge point of fingers are extracted as points of interest and matching is done utilizing the topological features. A new method is suggested to estimate the epipolar geometry between two uncalibrated cameras from stereo hand images. The fundamental matrix is estimated from uncalibrated stereo hand images. Fundamental matrix contains all geometric information that is necessary for establishing correspondence between two perspective images, from which 3D structure of an object can be inferred. The algorithm is tested with real calibrated and uncalibrated images. The experimental comparisons have shown the effectiveness and robustness of the method.

Statistical and Syntactic Analysis

[10] proposed a two level approach of statistical and syntactic analysis for the recognition of static and dynamic hand gestures respectively. The first level (statistical analysis) is based on Haar-like features and the AdaBoost learning algorithm. The second level (syntactic analysis) is based on a stochastic context-free grammar (SCFG). The Haar-like features effectively describe the hand posture pattern and the AdaBoost algorithm greatly speeds up performance and constructs a strong classifier by combining a sequence of weak classifiers. The posture detected by the first level are converted into a sequence of terminal strings according to the grammar. The SCFG algorithm analyzes the syntactic structure based on the detected postures. Based on the probability that is associated with each production rule, the gesture corresponding to an input string is identified by finding out the production rule that has the highest probability of generating the gesture.

Eigen Space Based Method

A hand gesture recognition method without any explicit feature detection step is proposed in [62]. The method suggests a novel eigenspace based framework to model dynamic hand gestures those incorporate both hand shape as well as trajectory information. In general, any feature-based method involves a separate time-consuming and noise-prone feature detection step which is avoided in [62]. The approach is useful in representing gestures that cannot be recognized by shape or from the trajectory information. Moreover the algorithm is immune to common hand shape

deformations: rotation, translation, scale and shear. Also the authors modeled an upper bound on success rate of a particular set of gestures by maximizing the distance between gesture classes in the eigenspace. The algorithm is tested using an eight class gesture set and it provided 100% classification of all the training and testing samples.

2.1.2 Hand Gesture Databases

The number of hand gesture databases available to the research community is limited [41]. As part of this dissertation two new hand posture databases namely NUS hand posture dataset-I (simple background) and NUS hand posture dataset-II (complex background) are developed. Table 2.4 lists the available hand gesture databases and sources.

2.1.3 Comparison of Methods

HMM based methods are effective and are widely used for hand gesture recognition. However HMM approaches have the disadvantage of a very elaborate training procedure to be effective. In contrast to the HMM based methods, the design of TDNN is attractive as its compact structure economizes on weights and makes it possible for the network to develop general feature detectors. Also, its hierarchy of delays optimizes these feature detectors by increasing their scope at each layer. Most importantly, its temporal integration at the output layer makes the network shift invariant (i.e., insensitive to the exact position of the gesture).

Graph based algorithms have the disadvantage of high computational

Table 2.3: Other methods for hand posture recognition: A comparison

Work	**S/D**	**Features used**	**Classification method**	**# classes**	**Accuracy (%)**	**Application**
Triesch *et al.* [97]	S	Gabor jets	EGM	12	92.9 (SB) & 85.8 (CB)	HRI
Ueda *et al.* [103]	S	joint angles	3D model fitting	4	NA	NA
Chen *et al.* [10]	S & D	haar-like features	Adaboost (S) / syntactic analysis based on SCFG (D)	4	NA	NA
Patwardhan *et al.* [62]	D	hand shape / trajectory (but no explicit feature detection step)	predictive eigen tracker	8	100	audio player control

S-static, **D**-dynamic, **EGM**-elastic graph matching, **SB**-simple background, **CB**-complex background, **HRI**-human robot interaction, **SCFG**-stochastic context-free grammar, NA-Not Applicable

Table 2.4: Hand gesture databases

Source	**Description**	**Work**
[40]	Two dynamic hand gesture datasets, one with 16 classes and another with 7 classes	[41]
Available on an e-mail request to yoonhs@etri.re.kr	48 class dynamic hand gestures representing alphanumeric characters & graphic elements	[117]
[95]	Two static hand gesture datasets, one with 10 classes (gray scale) and another with 12 classes (color). One dynamic hand gesture dataset with 4 classes	[96–98]
Available on an e-mail request to prahlad@nus.edu.sg	10 class simple background static hand gesture dataset	[77]
[1]	10 class complex background static hand gesture dataset	–

complexity, which leads to its unsuitability for real-time applications. However these methods are effective in the recognition of complex background gestures [97].

In general, appearance-based approaches have better real-time performance as the extraction of the 2-D image features that are employed is fast [10]. Appearance-based models lead to computationally efficient *purposive* approaches that work well under constrained situations, but seem to lack the generality desirable for HCI. Three-dimensional hand models offer a way of more elaborate modeling of hand gestures. Three-dimensional hand model-based approaches offer a rich description that potentially allows a wide class of hand gestures. However, 3-D models need a very large image database to cover all the characteristic shapes under different views. Matching the test image with all the images in the database is time consuming and computationally expensive which limits the usage of 3-D models for real-time applications.

2.2 Fuzzy-Rough Sets

This section provides an introduction to fuzzy-rough sets and a review of the classification and features selection algorithms in this field.

Real world classification problems often involve continuous data which makes the design of reliable classifiers difficult. One way to handle continuous data is by partitioning the data into crisp or discrete intervals. This process of discretization determines how coarsely the data is split into intervals. The crisp discretization is achieved by generating a set of

cuts of features within the dynamic ranges of the corresponding features. The positions of cuts are very sensitive to the subsets of the information system, which are used to generate the cuts, as well as to the methodology adopted. The position sensitivity of cuts may make the classification accuracy adversely affected. Fuzzy sets, which is a generalization of the classical sets, proposed by Zadeh in 1963 [118], offers solutions for tackling such difficulties associated with continuous data. It suggests the fuzzy discretization of the feature space and solves the problems associated with crisp discretization.

Z. Pawlak introduced the rough set theory in the early eighties, as a tool to handle inconsistencies among data [64–66]. A rough set is a formal approximation of a vague concept by a pair of precise concepts, the lower and upper approximations. No additional knowledge about the data, such as prior probability in probabilistic approach or grade of membership in fuzzy set theory, is required in rough sets. Rough sets handle uncertainty by computing the lower and upper approximations of the concept under consideration.

The concept of crisp equivalence class is the basis for rough set theory. A crisp equivalence class contains samples from different output classes. In addition, the various elements in an equivalent class may have different degrees of *belongingness* to the output classes. A combination of fuzzy and rough sets, namely fuzzy-rough sets [17, 18], is useful for decision making in such situations where both vagueness and indiscernibility are present. Fuzzy-rough set is a deviation of rough set theory in which the concept of crisp equivalence class is extended using fuzzy set theory to form fuzzy equivalence class [18]. A fuzzy similarity relation replaces an

equivalence relation in rough sets to form the fuzzy-rough sets. In fuzzy-rough sets the equivalence class is fuzzy in addition to the fuzziness of the output classes [84].

Let the equivalence classes be in the form of fuzzy clusters $F_1, F_2, ..., F_H$, which are generated by the fuzzy partitioning of the input set X into H number of clusters. Each fuzzy cluster represents an equivalence class and it contains patterns from different output classes. The definite and possible members of the output class are identified using lower and upper approximations [64] of the fuzzy equivalence classes. The description of a fuzzy set C_c (output class) by means of the fuzzy partitions under the form of lower and upper approximations $\underline{C_c}$ and $\overline{C_c}$ is as follows [84] :

$$\begin{aligned} \mu_{\underline{C_c}}(F_j) &= \inf_x\{\max(1-\mu_{F_j}(x), \mu_{C_c}(x))\} \quad \forall x \in X \\ \mu_{\overline{C_c}}(F_j) &= \sup_x\{\min(\mu_{F_j}(x), \mu_{C_c}(x))\} \quad \forall x \in X \end{aligned} \tag{2.1}$$

The tuple $\langle \overline{C_c}, \underline{C_c} \rangle$ is a fuzzy-rough set. $\mu_{F_j}(x)$ and $\mu_{C_c}(x)$ are fuzzy memberships of the sample x in the fuzzy equivalence class F_j and output class C_c respectively.

2.2.1 Feature Selection and Classification using Fuzzy-Rough Sets

Fuzzy and rough set theories are considered complementary in that they both deal with uncertainty: vagueness for fuzzy sets and indiscernibility for rough sets [18]. These two theories can be combined to form rough-fuzzy sets or fuzzy-rough sets [17, 18]. Combining the two theories provides the concepts of lower and upper approximations of fuzzy sets by similarity relations, which is useful for addressing classification problems.

The concept of fuzzy discretization of feature space for a rough set theoretic classifier is provided in [83]. The merit of fuzzy discretization over crisp discretization in terms of classification accuracy is demonstrated, when overlapping datasets are used. An entropy-based fuzzy-rough approach for extracting classification rules is proposed in [99]. A new fuzzification technique called Modified Minimization Entropy Principle Algorithm (MMEPA) is proposed to construct membership functions corresponding to the fuzzy sets.

The fuzzy-rough uncertainty is exploited in [84] to improve the classification efficiency of a conventional K-nearest neighbor (K-NN) classifier. The algorithm generalizes the conventional and fuzzy K-NN classifier algorithms. Another modification of the K-NN algorithm using fuzzy-rough sets is proposed in [106]. Fuzzy-rough concept is used to remove those training samples which are in the class boundary and overlapping regions. This improves classification accuracy. However the algorithm is applied to only one type of problem, the hand gesture recognition, whereas [84] reported more experimental results. [34] presents a new fuzzy-rough nearest neighbour (FRNN) classification algorithm, as an alternative to the fuzzy-rough ownership function (FRNN-O) approach in [84]. In contrast to [84], the algorithm proposed in [34] utilizes the nearest neighbors to construct the lower and upper approximations of the decision classes. The algorithm classifies test instances based on their membership to the lower and upper approximations. FRNN outperformed both FRNN-O and traditional fuzzy nearest neighbour (FNN) algorithm.

A new concept named as *consistence degree* is proposed in [120] to use as a critical value to reduce redundant attributes in a database. A rule

based classifier using a generalized fuzzy-rough set model is proposed. The classifier is effective on noisy data. A comparison between fuzzy-rough classifier and neural network (NN) classifier is provided in [39]. The fuzzy-rough classifier is reported as a better choice, as it needs lesser training time, has transparency, and has lesser dependance on the training data.

A feature selection method with fuzzy-rough approach and ant colony optimization is provided in [35]. Shen *et al.* [87] proposed a classifier that integrates a fuzzy rule induction algorithm with a rough set assisted feature reduction method. The classifier is tested on two problems, the urban water treatment plant problem and algae population estimation. Fuzzy-rough approach is utilized in [100] for decision table reduction. Unlike other feature selection methods, this method reduces the decision table in both vertical and horizontal directions (both the number of features and its dimensionality are reduced).

A robust feature evaluation and selection algorithm, using a new model of fuzzy-rough sets namely soft fuzzy-rough sets, is provided in [78]. This method is more effective in dealing with noisy data. [110] proposed a fuzzy-rough feature selection algorithm, with application to microarray based cancer classification. These works used standard classifiers (KNN, C5.0) for the classification process.

2.3 Biologically Inspired Features for Visual Pattern Recognition

The features of the image which are to be used for pattern recognition is an ongoing research topic in computer vision. Over the past three decades, there are lots of success stories in vision based pattern analysis. However, the mainstream computer vision has always been challenged by human vision, and the mechanism of human visual system is yet to be understood well, which is a challenge for both neuroscience and computer vision. The human visual system rapidly and effortlessly recognizes a large number of diverse objects in cluttered, natural scenes and identifies the specific patterns, which inspired the development of computational models of biological vision systems. Recent developments in the use of neurobiological models in computer vision tries to bridge the gap between neuroscience, computer vision and pattern recognition.

Hubel and Wiesel discovered the organization of receptive fields, and the properties of *simple* and *complex* cells in cat's primary visual cortex [31]. The cortical simple cell receptive fields are modeled [37] using a Gabor function (also known as Gabor wavelet or Gabor filter) which is described by the following equations.

$$F(x,y) = \exp\left(-\frac{({x_0}^2 + \gamma^2 y_0^2)}{2\sigma^2}\right) \times cos\left(\frac{2\pi}{\lambda}x_0\right), s.t. \tag{2.2}$$

$$x_0 = x\cos\theta + y\sin\theta \text{ and } y_0 = -x\sin\theta + y\cos\theta. \tag{2.3}$$

where,

- γ the spatial aspect ratio of the Gaussian function,
- σ the standard deviation of the Gaussian function,
- λ the wavelength of the sinusoidal term,
- θ the orientation of the Gaussian from the x-axis.

Gabor wavelet based features have good discriminative power between different textures and shapes in the image. Gabor filters resemble the receptive fields of neurons in the primary visual cortex of mammals [37]. Use of the 2D Gabor wavelet representation in computer vision was pioneered by Daugman [15].

Riesenhuber and Poggio extended this approach and proposed a hierarchical model of ventral visual object-processing stream in the visual cortex [81]. Serre *et al.* implemented a computational model of the system and used it for robust object recognition [85,86]. The features extracted by this model are known as the C_1 and C_2 standard model features (SMFs). The C_2 SMFs were later used for hand writing recognition [104] and face recognition [48]. These features are scale and position-tolerant, and the feature extraction algorithm does not require the segmentation of the image. Also the number of the extracted features is independent of the input image size. They used the features for robust object recognition [85, 86]. Later these features were used for hand writing recognition [104] and face recognition [48]. The algorithms proposed in this thesis utilize the C_2 features for the multi-class recognition of human faces and hand postures.

2.3.1 The Feature Extraction System

Table 2.5: Different layers in the C_2 feature extraction system.

Layer	**Process**	**Represents**
S_1	Gabor filtering	simple cells in V1
C_1	Local pooling	complex cells in V1
S_2	Radial basis functions	V4 & posterior inferotemporal cortex
C_2	Global pooling	inferotemporal cortex

The computational model proposed by Serre *et al.* consists of four layers (Table 2.5). Layer 1 (S_1) consists of a battery of Gabor filters with different orientations (4) and sizes (16 sizes divided into 8 bands). This imitates the simple cells in the primary visual cortex (V1) which filters the image for the detection of edges and bars. Layer 2 (C_1) models the complex cells in V1, by applying a MAX operator locally (over different scales and positions) to the first layer results. This operation provides tolerance to different object projection size, it's position and rotation in the 2-D plane of the visual field. In layer 3 (S_2), radial basis functions (RBFs) are used to imitate the V4 and posterior inferotemporal (PIT) cortex. This aids shape recognition by comparing the complex features at the output of C_1 stage (which corresponds to the retinal image) with patches of previously seen visual image and shape features (in human these patterns are stored in the synaptic weight of the neural cells). Finally, the fourth layer (C_2) applies a MAX operator (globally, over all scales and positions) to the output of layer S_2, resulting in a representation that expresses the best comparison with previously seen images. The output of layer 4 are the C_2 SMFs, which are used for the classification of the image.

Simple cells in the third layer implement an RBF, which combines bars and edges in the image to more complex shapes. RBFs are a major class of neural network model, comparing the distance between input and a prototype [5]. Each S_2 unit response depends in a Gaussian-like way on the Euclidean distance between a new input and a stored prototype. The prototype patches of different sizes (center of the RBF units) are drawn randomly (random image and position) from the training images at the

level of the second layer (C_1). Each patch contains all the four orientations. The third layer compares these patches by calculating the summed Euclidean distance between the patch and every possible crop (combining all orientation) from the image of similar size. This comparison is done separately with each scale-band representation in the second layer.

The final set of shift and scale invariant C_2 responses is computed by taking a global maximum over all scales and positions for each S_2 type, i.e., the value of the best match between a stored prototype and the input image is kept and the rest is discarded. Each C_2 feature corresponds to a specific prototype patch with a specific patch size (in layer 3). The more the number of extracted features the better is the classification accuracy. However when more number of features are extracted the computational burden (both for feature extraction and classification) will increase.

The feature selection and classification algorithms presented in Chapter 3 and 4 of the thesis extract the image features using the above feature extraction system (the focus of these chapters is on feature selection and classification aspects). The algorithms presented in Chapter 5 and 6 modify the feature extraction system for hand posture recognition application.

Chapter 3

Fuzzy-Rough Discriminative Feature Selection and Classification

Most of the fuzzy-rough classifiers surveyed in Chapter 2 (Section 2.2.1) are based on pre-defined fuzzy membership functions and are focussed either on classification or feature selection aspect. The fuzzy-rough approach is utilized as a preprocessing / supporting step. Direct construction of classifiers as an application of fuzzy-rough sets has been less studied [120]. The current work proposes a computationally efficient feature selection and classification algorithm for datasets with multidimensional feature space, utilizing the fuzzy-rough set approach. The algorithm automatically generates the fuzzy membership functions and selects the discriminative features in the dataset. The classification rules are generated using the selected subset of features, which further improves the computational efficiency of the classifier.

3.1 Feature Selection and Classification of Multi-feature Patterns

The computational expenses in any classification process are sensitive to the number of features used to construct the classifier. An abundance of features increases the size of the search space to be explored, thereby increasing the time needed for classification. The available features in a dataset can be categorized into four. 1) Predictive / relevant: the features which are good in the interclass discrimination, 2) Misleading: the features which affect the classification task negatively, 3) Irrelevant: the features which provide a neutral response to the classifier algorithm, and, 4) Redundant: the features of a class which has other relevant features for the discrimination. The presence of misleading features will reduce the classification accuracy and, the presence of irrelevant and redundant features will increase the computational burden. The removal of such features reduces the size of the rule base of a classifier by preserving the relevant and predictive features. This process is known as attribute reduction [67] or, in the context of machine learning, feature selection [93].

Similar and overlapped features in a dataset make the classification of patterns difficult. Interclass feature overlaps and similarities lead to indiscernibility and vagueness. Rough set theory [64, 65] is useful for decision making in situations where indiscernibility is present, and, fuzzy set theory [118] is suitable when vague decision boundaries exist.

This work considers the category of decision problems which is characterized with multidimensional feature space. Searching for an optimal feature subset in a high dimensional feature space is an NP complete

problem [2]. In the present work, the feature space is partitioned into fuzzy equivalence classes through fuzzy-discretization. The lower and upper approximations of the fuzzy equivalence classes are calculated for the training data set. The predictive features in the dataset are identified and if-then classification rules are generated using these features. The decision is made through a voting process using these rules. The proposed feature selection and classification algorithm has a polynomial time complexity. The algorithm selects the relevant features, and avoids the misleading, irrelevant and redundant ones. The selection of relevant features reduces the number of features required for classification, which further brings down the computational cost of the classifier.

Cancer and tumor classification using gene expression data is a typical multi-feature classification problem. Gene expression monitoring by DNA microarrays suggests a general strategy for predicting cancer classes, independent of previous biological knowledge [25]. However, the number of genes in the gene expression data is quite large (each gene profile is a feature utilized for the classification) and the availability of tissue samples / records is limited [71]. As the dataset has many more features than the number of available samples, the common statistical procedures such as global feature selection can lead to false discoveries [71]. These facts emphasize the need for a simple and robust classifier for such multi-feature classification problems.

According to Piatetsky-Shapiro *et al.* [71], the main types of data analysis needed for biomedical applications include, $a)$ Classification: classifying diseases or predicting treatment outcome based on gene expression

patterns, b) Gene selection: selecting the predictive features, and, c) Clustering: finding new biological classes or refining the existing ones. The first two are pattern recognition / data mining problems whereas the third requires domain knowledge in the biomedical field. The present work addresses the first two aspects. The proposed feature selection and classification algorithm is applied to five cancer / tumor datasets. The algorithm identified predictive genes and provided good classification accuracy for all the datasets considered.

The proposed algorithm is also tested on three visual pattern recognition tasks: hand posture recognition, human face recognition, and object recognition. The features of the images are extracted using a cortex-like mechanism [85], and, the classification is done using the proposed algorithm.

3.2 The Fuzzy-Rough Feature Selection and Classification Algorithm

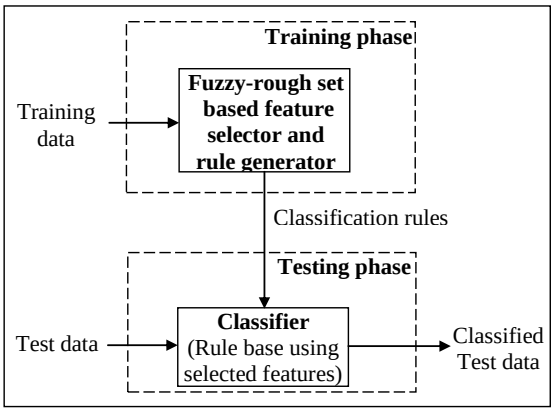


Figure 3.1: Overview of the classifier algorithm development.

The proposed fuzzy-rough set based feature selection and classification algorithm is discussed in this section. The aim is to come up with a classifier which identifies discriminative features in a dataset and classifies the data with less computational expense.

Fig. 3.1 shows the overview of the proposed classifier algorithm development process. The available dataset is divided into two sets: training and testing sets. In the training phase, the discriminative features in the data are selected and the classification rules are generated using a fuzzy-rough approach. The classifier developed is tested using the test data.

3.2.1 The Training Phase: Discriminative Feature Selection and Classifier Rules Generation

The training phase (Fig. 3.2) involves the fuzzy discretization of the feature space and, the formation of fuzzy membership functions using the cluster centers identified by the subtractive clustering technique [12]. The discriminative features in the dataset are selected and the classification rules are generated, using fuzzy lower and upper approximations of the fuzzified training data.

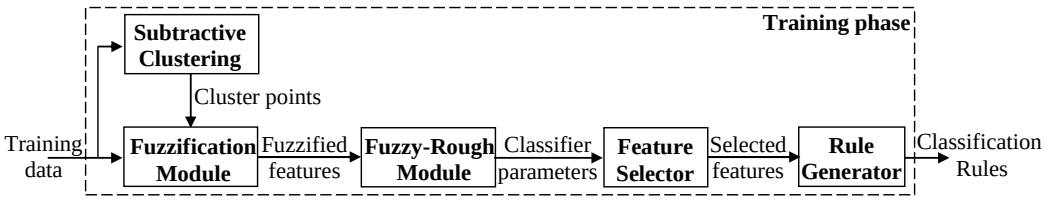


Figure 3.2: Training phase of the classifier.

Fuzzy Equivalence Classes and, the Lower and Upper Approximations

The fuzzy membership functions are formed using the feature cluster centers identified by the subtractive clustering technique [12]. Every data point is a potential cluster center. Subtractive clustering calculates a measure of the likelihood of a data point as a cluster center, based on the density of surrounding data points. The algorithm selects the data point with highest potential as the first cluster center and then removes all the data points in the vicinity (as specified by the subtractive clustering radius which usually lies within $[0.2, 0.5]$[1]) of the first cluster center. The second data cluster and its center point are identified next. This process is repeated until every data sample lies within the radius of one of the cluster centers.

The concept behind the proposed algorithm is explicated with the following example. Consider a 3 class classification problem with a 2 dimensional feature space. Let the sample distribution in the feature space $A - B$ is as shown in Fig. 3.3(a) and the output class considered is class 2. The fuzzy membership function is centered at the cluster center of feature A (of class 2). The minimum and maximum values of feature A (of class 2) position the left and right sides of the membership function.

In the example, the fuzzy membership function forms an equivalence class. The samples near to the cluster center have maximum membership in class 2, as these samples have a better chance to be in class 2.

[1]The subtractive clustering radius represents the width of the data considered in each step of the clustering. A radius within [0.2 0.5] leads to diameter within [0.4 1.0] and it covers 40% to 100% of the data width. This is the usual range reported in the literature [12]. The number of rules and accuracy decrease with the radius.

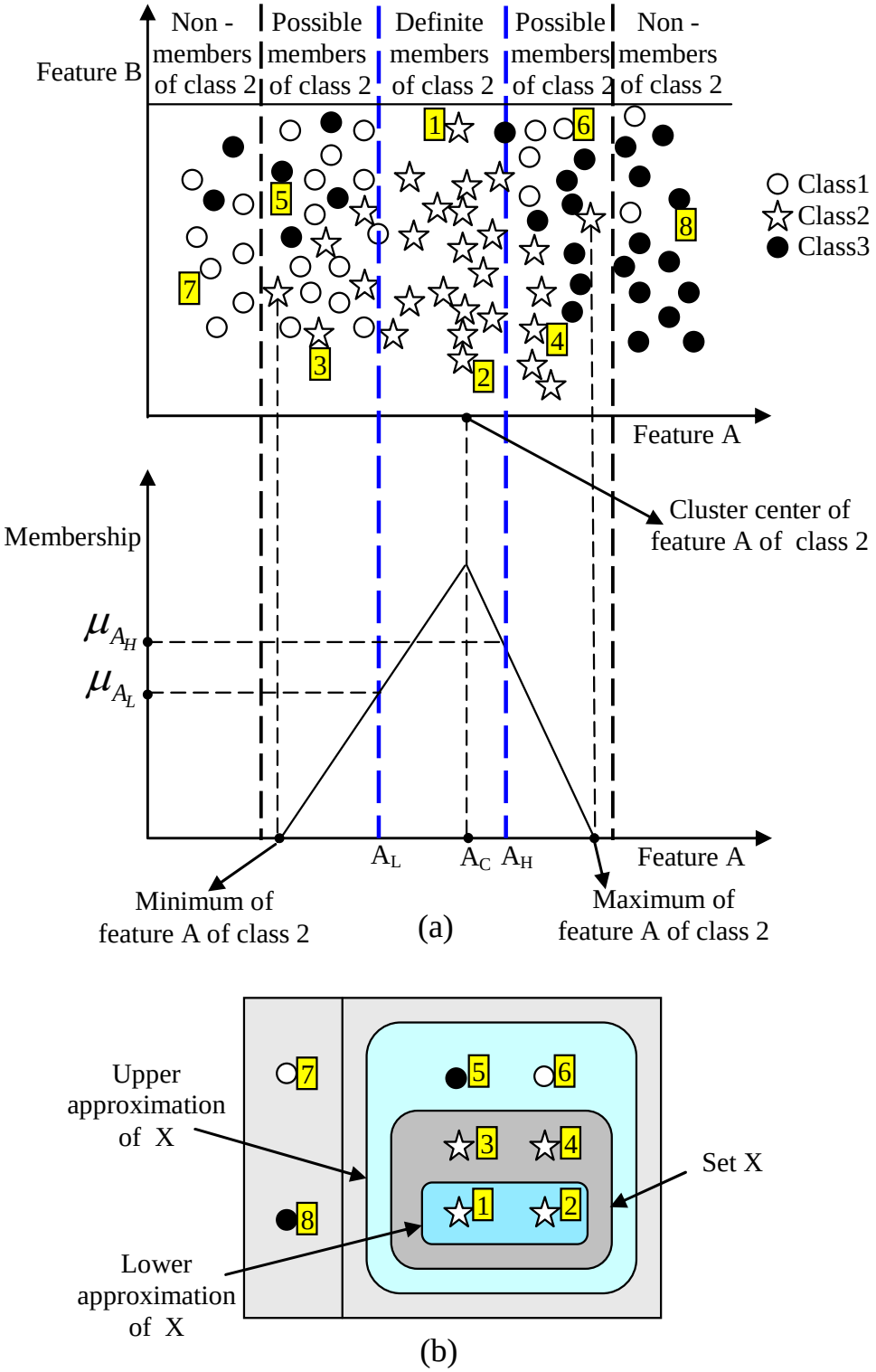


Figure 3.3: (a) Feature partitioning and formation of membership functions from cluster center points in the case of a 3 class dataset. The output class considered is class 2. (b) Lower and upper approximations of the set X which contains samples 1-8 in (a).

However the same fuzzy equivalence class contains samples from different output classes which leads to fuzzy-rough uncertainty. The proposed algorithm identifies the membership values μ_{A_L} and μ_{A_H} (3.3) that partition the definite and possible members of the output class (class 2 in this example, Fig. 3.3(a)) and identifies the relevant features for discriminating the output class .

Let X be the set of samples labeled 1-8 in Fig. 3.3(a). The set contains samples from different partitions of the feature space. The lower and upper approximations of the set X are shown in Fig. 3.3(b). The lower approximation consists the definite members and the upper approximation consists the definite and possible members of class 2.

The cluster centers are identified and the membership functions are formed for each feature of every class. The training data is then fuzzified using the generated membership functions and, the lower and upper approximations are obtained. The set of membership values $\{\mu_{A_L}, \mu_{A_H}\}$ and feature values $\{A_L, A_H\}$ that partition the definite and possible members are utilized to identify the discriminative features, and to generate the classification rules respectively[2].

The distribution of data samples in the feature space may have varying amount of overlaps between different classes, along various feature axes. The proposed algorithm identifies the discriminative features, the features which have minimum interclass overlap, as the predictive/ relevant features in the dataset, and generates the classification rules using them.

[2]Refer Appendix-A for an illustration of the formation of fuzzy membership functions, and the calculation of $\{\mu_{A_L}, \mu_{A_H}\}$ and $\{A_L, A_H\}$.

All the datasets considered in this paper have large number of features. The presence of large number of features increases the possibility of identifying predictive features with less interclass overlap.

Feature Selection

The fuzzy equivalence class can contain samples from different output classes. Let μ_{A_L} and μ_{A_H} be the membership values that partition the definite and possible members of an output class (Fig. 3.3(a)). μ_{A_L} and μ_{A_H} are calculated as follows.

Let MF be the fuzzy set associated with a particular feature in a class and,

$$\mu^{max}(C_i) = \max\{\mu_{MF}[A_{C_i}(l)]\}, \tag{3.1}$$

where,

$A_{C_i}(l)$ represents the feature values of the samples from the class C_i,

and,

$$C_{max} = \text{argmax}_{C_i}\{\mu^{max}(C_i)\}, \tag{3.2}$$

then,

$$\begin{aligned} \mu_{A_L} &= \max_{C_i \neq C_{max}, A_{min} \leq A \leq A_C} \{\mu^{max}(C_i)\}, \\ \mu_{A_H} &= \max_{C_i \neq C_{max}, A_C \leq A \leq A_{max}} \{\mu^{max}(C_i)\}. \end{aligned} \tag{3.3}$$

μ_{A_L} and μ_{A_H} are the maxima of the membership values associated with

data samples belonging to classes other than C_{max}. Once these membership values are calculated, the features of the data are sorted in descending order of d_μ where d_μ is the average value[3] of $d_{\mu_{A_L}}$ and $d_{\mu_{A_H}}$ (Fig. 3.4).

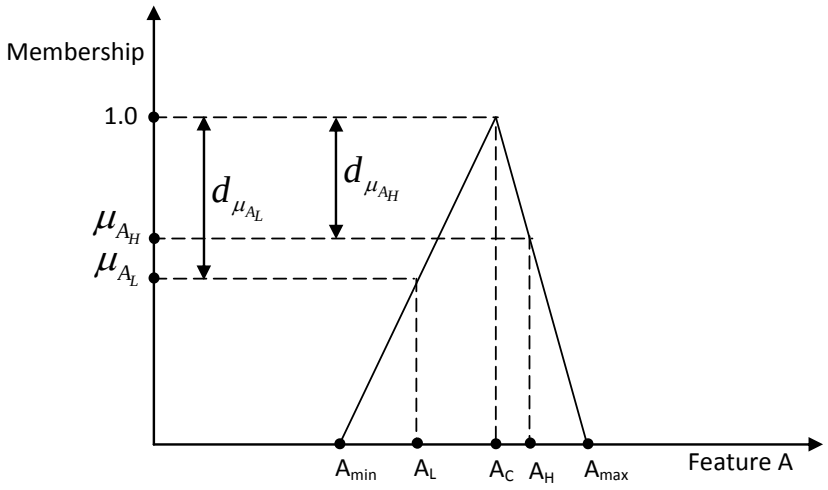


Figure 3.4: Calculation of d_μ.

$$d_\mu = (d_{\mu_{A_L}} + d_{\mu_{A_H}})/2, \tag{3.4}$$

where,

$$d_{\mu_{A_L}} = 1 - \mu_{A_L} \text{ and } d_{\mu_{A_H}} = 1 - \mu_{A_H} \tag{3.5}$$

The value of d_μ is an indication of the discriminative ability of a particular feature. A high value of d_μ indicates that the corresponding feature is good for interclass discrimination, as it has less interclass overlap. $d_\mu = 0$ ($\mu_{A_L} = \mu_{A_H} = 1$) represents an indiscriminating feature.

The classification rules are generated using the first n features from the sorted list (using the features which provide higher values of d_μ). The

[3]The algorithm provides better results when the average value is considered. The other possible values for d_μ are $d_{\mu_{A_L}}$, $d_{\mu_{A_H}}$ or a weighted average of $d_{\mu_{A_L}}$ and $d_{\mu_{A_H}}$.

proposed algorithm is tested by varying the number of selected features n. The classification accuracy of the algorithm first increases and then saturates with respect to n, for all the datasets considered.

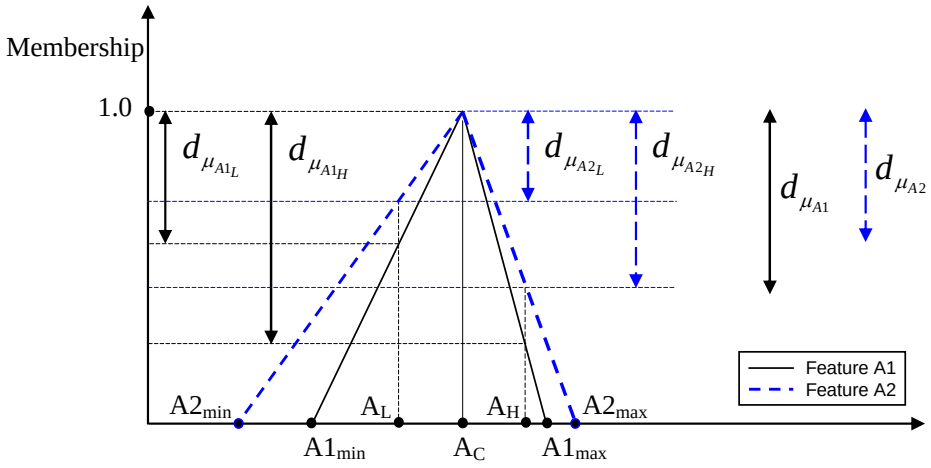


Figure 3.5: Calculation and comparison of d_μ for two features A1 and A2 with different feature ranges.

The capability of d_μ in representing the relevance of features is explained as follows. The calculation of d_μ for two features A1 and A2 is depicted in Fig. 3.5. The range of feature A1 ($A1_{max} - A1_{min}$) is less than that of A2 ($A2_{max} - A2_{min}$). Let the feature range $[A_L, A_H]$ and feature cluster center A_C are the same for the two features A1 and A2. In this case the feature selection algorithm provides preference to feature A1, as it has a denser sample distribution. The algorithm provides higher value for $d_{\mu_{A1}}$ (the average of $d_{\mu_{A1_L}}$ and $d_{\mu_{A1_H}}$) than $d_{\mu_{A2}}$. Within the feature range of A1, $A1_{max} - A_C < A_C - A1_{min}$, which implies that the distribution of samples is sparser within [$A1_{min}$, A_C] and denser within [A_C, $A1_{max}$] (as A_C is the feature cluster center). The proposed algorithm provides preference to the denser feature range $[A_C, A_H]$ and assigns a high value to

the d_μ corresponding to it ($d_{\mu_{A1_H}}$), compared to the d_μ of the sparse feature range $[A_L, A_C]$ ($d_{\mu_{A1_L}}$).

Classification

The feature values A_L and A_H that partition the lower and upper approximations of the fuzzy equivalence class entails the rule: if the value of a feature A is within (A_L, A_H) then the sample belongs to the class C_{max} (Fig. 3.4, A.1). This feature range decides whether a particular sample is a definite member of the output class C_{max}. The rule always holds true for the training samples. However some of the rules classify only a small number of training samples (say 1 or 2), if the samples from various classes are well mixed. To increase the reliability of the classifier, only those rules which classify two or more number of training samples are stored in the rule base.

In order to classify new samples, the classification rules are generalized as follows (*Rule 1* & *2*).

Let $\{A_{L_{ij}}, A_{H_{ij}}\}$ be the set of feature values obtained as per (3.3) where,
i 1,. . . , p : p - number of classes,
j 1,. . . , q' : q' - number of selected features,

then the samples are classified using the following two rules. *Rule 1* is a voting step whereas *Rule 2* is the decision maker.

Rule 1:

$$IF\ [A_{L_{ij}} < A_j < A_{H_{ij}}]\ THEN\ [N_{C_i} = N_{C_i} + 1] \tag{3.6}$$

where,

A_j j^{th} feature of the sample to be classified,

N_{C_i} the number of votes for a particular class C_i.

Rule 2:

$$C = argmax_{C_i}\{N_{C_i}\} \tag{3.7}$$

where,

C is the class to which the sample belongs (the output class).

A detailed flowchart of the training phase of the classifier is provided in Fig. 3.6.

The proposed classifier is a *margin classifier* that provides the minimum distance from the classification boundary, namely *margin of classification*, for each sample. The margin of classification for the proposed classifier is defined as,

$$MC = N_p - N_{nMAX} \tag{3.8}$$

where,

MC is the margin of classification,

N_p is the number of positive votes, and

N_{nMAX} is the maximum number of negative votes.

In case of a dataset with three classes, the number of votes N_{C_1}, N_{C_2}, and N_{C_3} are calculated for a sample under consideration (*Rule 1*). *Rule 2* then identifies the class with maximum votes. *Rules 1* and *2* serve to form the classifier rule base, keeping the algorithm computationally simple.

For a sample from class 1, let the values $N_{C_1} = 90$, $N_{C_2} = 5$, and

$N_{C_3} = 10$. Then the ***MC*** for the sample is $90 - 10 = 80$. In this case the sample received 90 positive votes.[4] The number of negative votes are 5 and 10 in classes 2 and 3 respectively. A positive margin indicates correct classification whereas negative margin indicates misclassification. The average margin of classification for a dataset indicates the discriminative power of the classifier for that dataset. The experimental results (Section 3.3) evidence good discriminative power of the proposed classifier for all the datasets considered.

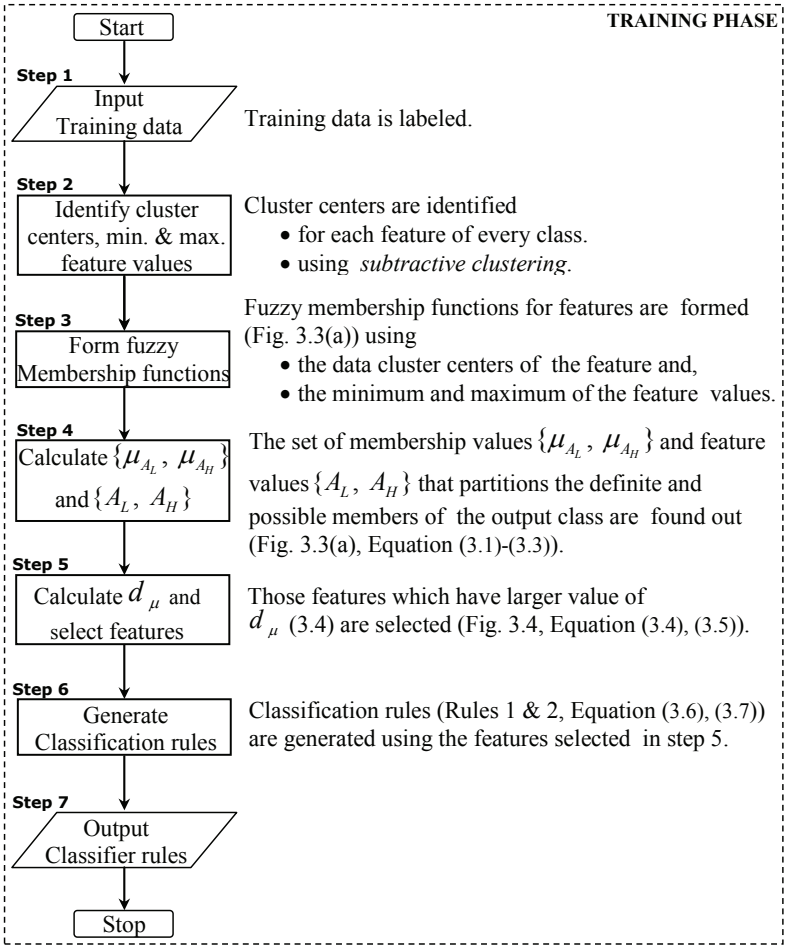


Figure 3.6: Flowchart of the training phase.

[4]Voting is positive if the voted class and the actual class are the same. Otherwise it is negative.

3.2.2 The Testing Phase: The Classifier

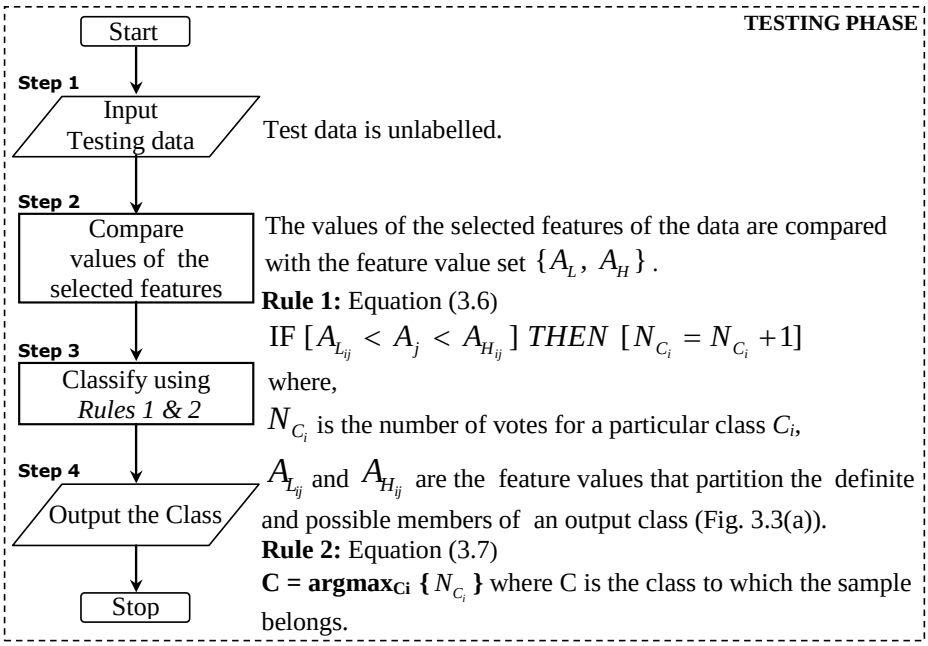


Figure 3.7: Flowchart of the testing phase.

The fuzzy-rough classifier is formed using the classification rules generated in the training phase. Fig. 3.7 shows the flowchart of the testing phase (the classifier). The selected features of a test sample are compared with the feature values A_L and A_H, and the classification is done using *Rules 1* and *2*. Each of the execution steps of the classifier is a direct comparison of feature values, using the classification rules, which makes the algorithm computationally simple. The classification results are discussed in Section 3.3.

3.2.3 Computational Complexity Analysis

This section provides a detailed computational complexity analysis of the training and the testing (the classifier) algorithm. Both training and testing algorithms have polynomial time complexity.

Computational Complexity of the Classifier Training Algorithm

```
read {train_data} ; % read all the features of the samples
for i = number_of_classes
   for j = number_of_features;
      calculate {Ac} ; % finding feature cluster centers - using subtractive clustering
      calculate {Amin}, {Amax}; % finding min. & max. values of features
      calculate { μ }; % calculation of fuzzy memberships in each class
      calculate { μ_AL , μ_AH }; % Equation (3.1)-(3.3)
      calculate { A_L , A_H }; % finding features values that partitions definite
                               % and possible members of an output class
      calculate { d_μ }; % Equation (3.4), (3.5)
   end
end
for i = number_of_classes
     Asort = sort { d_μ }; % features are sorted in the descending order of d_μ
                           % - for the feature selection
end
write { A_L , A_H }, {Asort}; % store the parameters of the rule base
```

Figure 3.8: Pseudo code of the classifier training algorithm.

Fig. 3.8 shows the pseudo code of the classifier training algorithm. The different parameters at the input of the training algorithm are the number of classes, the number of features, and the number of samples per

class. Let p be the number of classes, q be the number of features, and r be the number of samples per class. The complexity of the algorithm is as follows : $O(pqr)$ for reading the training data, $O(pqr^2)$ for finding the cluster centers (using subtractive clustering), $O(pqr)$ for finding the minimum and maximum of feature values, $O(p^2qr)$ for calculating the fuzzy memberships, $O(p^2qr)$ for finding the membership values μ_{A_L} and μ_{A_H}, $O(pq)$ for finding feature values A_L and A_H, $O(pq)$ for calculating d_μ, $O(pq\ log(q))$ for sorting d_μ, and, $O(pq)$ for storing the rule base parameters. The overall complexity of the algorithm is $O(pqr^2) + O(p^2qr) + O(pq\ log(q))$, which is polynomial time.

Computational Complexity of the Classifier

```
read { A_L , A_H }, {A_sort}; % read the parameters of the rule base
read {test_data} ; % read the selected n features of the sample - corresponding
                   % to the first n feature indices listed in {A_sort}
for i = number_of_classes
  for j = number_of_features; % selected n number of features
    IF [A_L < A < A_H] % compare the features of the test sample with
                       % feature value set {A_L, A_H}
      vote(i) = vote(i) +1; % Rule 1, Equation (3.6)
    end
  end
end
class = index {max (vote)} ; % Rule 2, Equation (3.7), find the class index
                             % which received maximum votes
print {class} ;
```

Figure 3.9: Pseudo code of the classifier.

The pseudo code of the classifier is shown in Fig. 3.9. The different

parameters at the input of the classifier are the number of classes and the number of selected features. Let p be the number of classes and q' be the number of selected features. The complexity of the algorithm is as follows : $O(pq')$ for reading the classifier rule parameters, $O(q')$ for reading the selected features of the sample, $O(pq')$ for the voting process, and, $O(p)$ for finding the class index which received the maximum votes. The overall complexity of the proposed classifier algorithm is polynomial time, $O(pq')$.

3.3 Performance Evaluation and Discussion

The proposed feature selection and classification algorithm is tested using 5 cancer (Table 3.1) and 3 image datasets (Table 3.2). The reported results are the classification accuracy, the number of features used per class, the total number of features used (which is less than or equal to the product of number of features used per class and the number of classes), and the average margin of classification (MC) (Table 3.3 & 3.5). The variation in classification accuracy with respect to number of selected features is reported (Fig. 3.10). A comparison of the classification accuracy of the proposed algorithm with relevant classification methods is also provided.

Table 3.1: Details of cancer datasets

Dataset	# Classes	# Samples
Leukemia [24]	3	72
Tumor vs. normal samples [24]	2	75
Lung cancer [27]	2	181
Small round blue cell tumor [42]	4	83
Central nervous system embryonal tumor [24]	5	42

3.3.1 Cancer Classification

Cancer classification, which is a typical multi-feature classification problem, is based on microarray based gene expression data. Accurate classification of cancer is necessary for diagnosis and treatment. As the number of available samples is limited, 10 fold cross-validation is done for all cancer datasets, except for the fifth dataset (central nervous system embryonal tumor), for which *leave one out cross validation* is done. The classification results are compared with that of Support Vector Machines (SVM) implemented using LIBSVM [7] (Table 3.3). The classification results are also compared with those reported in the literature (for this comparison, the training and testing of the algorithm are done using the same sample divisions as that in the compared work) (Table 3.4).

Leukemia Classification

The leukemia dataset [24] consists of a total of 72 samples. Each sample has 7,129 gene expression profiles and each gene profile is a feature in the classification process. Originally the dataset was created and analyzed for the binary classification into acute lymphoblastic leukemia (ALL) and acute myeloblastic leukemia (AML) [25]. Jirapech-Umpai *et al.* [36] separated the dataset into three classes by using subtypes of ALL. Seventy two samples in the dataset are divided into three classes: ALL B-cell (38), ALL T- cell (9), and AML (25). In the present work, the three class classification is carried out and the 55 top ranked genes[5] by the RankGene method [36] are utilized for classification. For the 10 fold cross validation,

[5]The list of 55 top ranked genes is available in [36].

4 samples of ALL B-cell, 1 sample of ALL T-cell and 3 samples of the AML are considered in one subset (some samples are repeated in the subsets, as 10 subsets each having 8 samples are formed using the 72 samples). The outcome of the classification is provided in Table 3.3. The proposed algorithm provided the maximum classification accuracy of 100%, when a total of 35 features are used. The variation in classification accuracy with respect to the number of selected features per class is shown in Fig. 3.10(a). The SVM classifier provided 94.44% accuracy for this dataset (even though all available features are used in SVM).

The classification of the same dataset is done in [36] using evolutionary algorithm and GA-KNN classifier which reported 98.24% accuracy. In their work, 38 samples are used for training and 34 samples are used for testing. The proposed algorithm is tested using the same sample divisions and provided 100% classification accuracy (Table 3.4). Fifty gene profiles (features) are used in [36], whereas 35 gene profiles (15 selected features per class) are used by the proposed algorithm.

Tumor vs. Normal Sample Classification

In [54], tumor detection is done using MicroRNA expression profiles. The dataset [24] consists of expression profiles of tumor and normal samples of multiple human cancers. Each sample has 217 expression profiles. A k-Nearest Neighbor (kNN) classifier is built and trained using the human tumor / normal samples and it is utilized for the prediction of tumor in mouse lung samples [54]. Also the algorithm identified markers, the features that best distinguishes tumor and normal samples. The training data consists of colon, kidney, prostate, uterus, lung, and breast human

tumor / normal samples (43 tumor and 32 normal samples) and the testing is done using mouse lung tumor / normal samples (12 samples). The kNN classifier provided 100% correct detection of tumor (Table 3.4), when 131 markers are used [54].

As this is a tumor detection problem (rather than a classification problem), the lower and upper approximations, and the corresponding membership (μ_{A_L}, μ_{A_H}) and feature values (A_L, A_H) are calculated only for the tumor samples (and not for the normal samples). The presence of cancerous tumor in the test sample is predicted if more than 50% of the selected features vote the sample as tumor sample. The algorithm provided 100% correct detection of tumor in the mouse lung samples, when 35 markers (20 selected features per class) are used.

In order to substantiate the reliability of the classifier, an additional 10 fold cross test is done using the human tumor / normal samples (43 tumor and 32 normal samples). Each of the subsets consists of 5 tumor and 4 normal samples (some samples are repeated in the subsets, as 10 subsets each having 9 samples are formed using the 75 samples). The proposed algorithm provided 96.67% classification accuracy when a total of 35 selected features are used (Table 3.3). Fig. 3.10(a) shows the variation in classification accuracy with respect to the number of selected features per class. For this dataset the SVM classifier provided an accuracy of 94.66%.

Lung Cancer Classification

Lung cancer dataset [27] is used by Gordon *et al.* [28] for gene expression ratio based diagnosis of mesothelioma (MPM) and adenocarcinoma

(ADCA) of the lung. The set contains gene expression profile of 181 tissue samples (31 MPM and 150 ADCA). For the lung cancer dataset the number of genes is 12,533 and all of them are considered for the classification. Three MPM and 15 ADCA samples are used in one subset to do the 10 fold cross validation (One subset contains 18 samples. The execution is repeated until all the 181 samples are tested). The proposed algorithm provided 100% classification accuracy (Table 3.3) when a total of 44 features (40 selected features per class) are used. The variation in classification accuracy with respect to the number of selected features per class is shown in Fig. 3.10(b). For this dataset the SVM classifier also provided 100% accuracy.

In [28], 99% classification accuracy is achieved for the same dataset, when 16 MPM and 16 ADCA samples are used for training, whereas the proposed algorithm provided 100% accuracy, when trained in a similar manner (Table 3.4). However, only 6 genes are used in [28] as the method is based on the ratio of gene expressions, whereas the proposed algorithm needs 24 genes to provide 99% accuracy and 44 genes to provide 100% accuracy.

Small Round Blue Cell Tumor Classification

The dataset used is NCI's dataset [42] of small round blue cell tumors (SRBCTs) of childhood [43]. There are four classes : Burkitt lymphoma (BL), Ewing sarcoma (EWS), Neuro blastoma (NB) and Rhabdomyosarcoma (RMS). A total of 83 samples are provided (11 BL, 29 EWS, 18 NB and 25 RMS). The dataset consists of 2,308 genes whereas the present

work utilizes the top 200 individually discriminatory genes (IDGs)[6] identified by Xuan *et al.* [111]. For the 10 fold cross test 1 BL, 3 EWS, 2 NB and 2 RMS are considered in one subset (One subset contains 8 samples. The execution is repeated until all the 83 samples are tested). The classification results are provided in Table 3.3. The proposed algorithm provided an accuracy of 98.75%, when a total of 103 features are used. Fig. 3.10(b) shows the variation in classification accuracy with respect to the number of selected features per class. For this dataset the SVM provided similar performance as that of the proposed algorithm. The accuracy provided by SVM classifier is 98.8%.

The classification of the same dataset is done in [43], using neural network (NN) classifier with 96 features, and in [111], using multilayer perceptron (MLP) classifier with 9 features. They used 63 samples for training and 20 samples for testing, and achieved 100% and 96.9% classification accuracies respectively. The proposed algorithm provided an accuracy of 100% when the same training and testing samples are used, with 103 features (35 selected features per class). The proposed classifier provided better accuracy compared to the MLP classifier. However, in the MLP classifier, the classification is done with less number of genes (only 9) as Jointly Discriminatory Genes (JDGs) [111] are used.

Central Nervous System Embryonal Tumor Classification

The gene expression based prediction of central nervous system embryonal tumor is reported in [73]. The multiple tumor classes are predicted using k Nearest Neighbor (kNN) algorithm. There are five classes namely

[6]These genes are listed in the additional material (Table S1) of [111].

medulloblastoma (10 samples), malignant glioma (10 samples), AT/RT (an aggressive cancer of the central nervous system, kidney, or liver that occurs in very young children) (10 samples), normal cerebellum (4 samples), and supratentorial PNETs (primitive neuroectodermal tumor) (8 samples). A total of 42 samples are provided, each having 7129 gene profiles. The evaluation method used is *leave one out cross validation*, similar to [73]. The accuracy achieved by the proposed algorithm is 85.71 %, whereas that achieved by the kNN classifier is 83.33% (Table 3.3). The proposed algorithm used 90 selected genes (markers) per class whereas [73] used only 10 markers per class (the proposed algorithm provided a lesser accuracy when only 10 markers per class are used - Fig. 3.10(c)). The accuracy provided by the SVM classifier for this dataset is 80.95%.

3.3.2 Image Pattern Recognition

The image based pattern classification problems considered are hand posture recognition, face recognition, and object recognition. Image features used are C_2 standard model features (SMFs) [85], which are extracted using a computational model of the ventral stream of visual cortex [85, 86]. A total of 1000 C_2 SMFs are extracted. Two fold cross validation is done for hand posture and face datsets. For the object dataset, 100 random images per class are used for training, and the testing is done using all the remaining images. The reported classification accuracy is the average over 10 such runs. The classification results for image datasets are compared with that of SVM and principal component analysis (PCA-eigenface method [102]), both implemented using the C_2 SMFs (Table 3.5).

Table 3.2: Details of hand posture, face and object datasets

Dataset	# Classes	# Samples
Jochen Triesch hand posture dataset [98]	10	240
A subset of Yale face dataset [23]	10	640
Caltech object database [21]	4	3479

Table 3.3: Summary and comparison of cross validation test results - Cancer datasets (Training and testing are done by cross validation)

Dataset	**Accuracy (%) -proposed method**	**# features / class**	**Total # features**	**Average MC**	**Accuracy (%) -SVM**
Leukemia	100	15	35	7.03	94.44
Tumor vs. normal samples data	96.67	20	35	-	94.66
Lung cancer	100	40	44	22.44	100
Small round blue cell tumor	98.75	35	103	11	98.80
Central nervous system embryonal tumor	85.71	90	433	35	80.95

Table 3.4: Comparison of classification accuracy (%) with reported results in the literature - Cancer datasets (Training and testing are done using the same sample divisions as that in the compared work)

Dataset	**Proposed method**	**Benchmark**
Leukemia	100	98.24 [36]
Tumor vs. normal samples data	100	100 [54]
Lung cancer	100	99 [28]
Small round blue cell tumor	100	100 [43]
Central nervous system embryonal tumor	85.71	83.33 [73]

Hand Posture Recognition

Hand posture dataset considered is Jochen-Triesch hand posture dataset [95]. Ten classes of hand postures, performed with 24 different persons, against light background, are considered for the classification. The images vary in size of the hand and shape of the postures. A total of 240 samples, with 24 images from each class, are provided. The algorithm is tested by dividing the dataset equally into training and testing samples (12 images from each class). The algorithm is repeated by interchanging training and testing sets and the average results are reported (two fold cross test).

The proposed algorithm provided the maximum classification accuracy of 100%, whereas PCA provided an accuracy of 98.75% (Table 3.5). The SVM classifier also provided 100% accuracy. However, all the 1000 features are used for the classification using PCA and SVM, whereas the proposed algorithm used only 178 features (20 selected features per class). The variation in classification accuracy with the number of selected features per class is shown in Fig. 3.10(d). The accuracy saturates at 100% when 20 or more selected features per class are used.

Face Recognition

Face dataset considered is a subset of the Yale face database B [23], which contains 10 classes of face images, taken from different lighting directions. It consists of 640 frontal face images, 64 from each class. The algorithm is tested in a similar manner as done for the hand posture dataset. The proposed algorithm as well as the SVM classifier provided the maximum

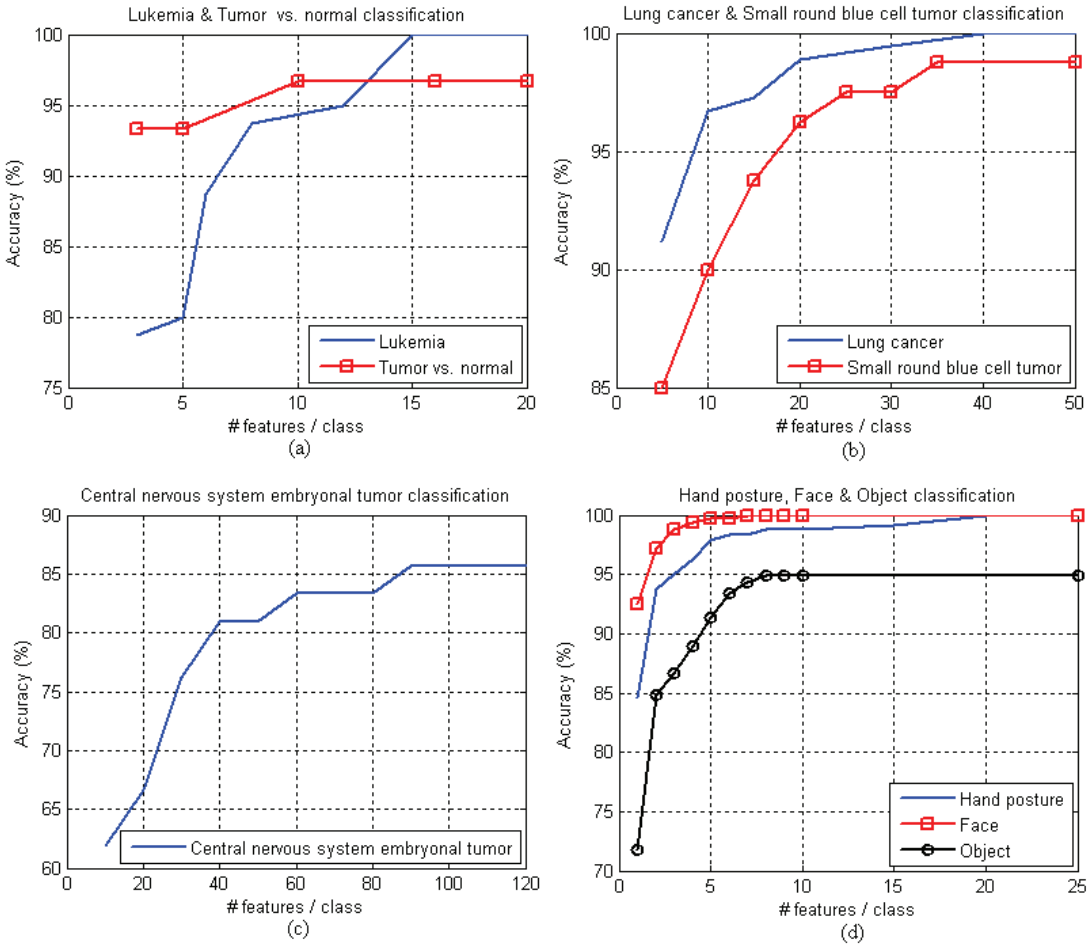


Figure 3.10: Variation in classification accuracy with the number of selected features.

classification accuracy of 100%, whereas the PCA provided 99.7% accuracy (Table 3.5). The whole set of features (1000) is used in PCA and SVM. The proposed algorithm is tested by varying the number of selected features (Fig. 3.10(d)). The classification accuracy saturates at 100% when only 7 selected features per class (a total of 56 features) are used.

Object Recognition

The classes considered in object recognition are human frontal face, motorcycle, rear car, and airplane [21]. A total of 3479 images (450 faces,

Table 3.5: Summary and comparison of cross validation test results - hand posture, face and object recognition

Dataset	Accuracy (%) -proposed method	# features / class	Total # features	Average MC	Accuracy (%) -PCA	Accuracy (%) -SVM
Hand posture dataset	100	20	178	12.65	98.75	100
Face dataset	100	7	56	5.68	99.7	100
Object dataset	94.96	8	30	3.39	88.37	94.84

800 motor cycles, 1155 cars, and 1074 air-planes) are provided. The training set consists of 100 randomly selected images from each class, and the testing is done using remaining 3079 images. The reported results are the average over ten such runs. The proposed algorithm provided an average classification accuracy of 94.96%, whereas PCA provided 88.37% accuracy (Table 3.5). The SVM classifier provided an accuracy of 94.84%, equivalent to that of the proposed algorithm. All the 1000 features are used in PCA and SVM, whereas the proposed algorithm needs only 30 features (8 selected features per class). The recognition accuracy saturates at 94.96% when 8 or more selected features per class are used (Fig. 3.10(d)).

3.4 Summary

A feature selection and classification algorithm based on the concept of fuzzy-rough sets is proposed. The fuzzy membership functions, which partition the feature space into fuzzy equivalence classes, are evolved automatically from the training dataset. The fuzzy membership values that partition the lower and upper approximations of the fuzzy equivalence

classes are utilized to identify the discriminative features in the dataset. The classifier rules are generated using the identified predictive features and the samples are classified through a voting process using these rules. A measure of the quality of classification, the *margin of classification*, is defined for the proposed classifier.

The performance of the algorithm is evaluated with two types of multiple feature datasets namely cancer and image datasets, and compared with relevant classification techniques. The proposed algorithm provided classification accuracy which is equivalent or better than that provided by the compared methods, with less computational efforts, and with good margin of classification.

Selection of relevant features reduced the number of features required for classification, which in turn reduced the computational burden of the classifier. The classification accuracy first increases and then saturates with respect to the number of selected features, for all the datasets considered. The effectiveness of the classifier in different types classification problems proves its generality. The time needed for rule generation as well as classification is several seconds, whereas that of the conventional machine learning algorithms is of the order of minutes or even hours. The proposed classifier is effective in cancer and tumor classification, which has high impact in the biomedical field. Also the effectiveness of the algorithm in image pattern recognition is evident, which is useful in human-computer interaction, human-robot interaction, and virtual reality.

In this work the first (main) cluster center of a feature is considered for the generation of fuzzy membership functions. Additional membership

functions (and classification rules) can be generated by considering the second and third cluster centers, which can improve the performance of the algorithm. The feature selection algorithm needs modification accordingly, as the density of cluster centers vary (in general, first cluster center is denser than the second, which in turn is denser than the third). This modification is carried out and the results are presented in Chapter 4.

Chapter 4

Hand Posture and Face Recognition using a Fuzzy-Rough Approach

4.1 Introduction

The multi-class recognition of hand postures and human faces is considered in this chapter. A fuzzy-rough set based novel classification algorithm that classifies the images at a less computational cost is proposed. The features of the images are extracted using a cortex-like mechanism [85]. The algorithm then partitions the feature space by fuzzy-discretization into fuzzy equivalence classes. The lower and upper approximations of the fuzzy equivalence classes are obtained which are used to derive the fuzzy if-then rules. These rules are utilized for classifying a new image, through a voting process. The proposed algorithm has a polynomial time complexity.

By identifying the decision attributes, the performance of a classification algorithm can be improved. The cost of classification is sensitive to

the number of attributes / features used to construct the classifier. In the present work, Genetic Algorithm (GA) is utilized to reduce the number of features required for classification, by identifying and removing irrelevant and redundant features. The proposed classifier is a margin classifier and it provides *margin of classification*, a measure of the distance from the classification boundary. The margin of classification is used to define the fitness function. The proposed fitness function assists in the feature selection process without compromising on the classification accuracy and margin. The reduction in number of features needed for classification reduced the number of features to be extracted. This increased the speed of both feature extraction as well as classification processes, which makes the proposed algorithm suitable for real-time applications.

The proposed algorithm is tested using three face datasets (a subset of Yale face database B [23], a subset of color FERET database [69], and a subset of CMU face dataset [90]) and two hand posture datasets (a newly created dataset namely NUS hand posture dataset, and Jochen Triesch hand posture dataset [95]). Also the algorithm is tested by capturing the hand images online, and the performance is compared with that of Support Vector Machines (SVM).

4.2 The Fuzzy-Rough Classifier

The proposed recognition algorithm is discussed in this section. The recognition of a pattern in the image consists of two processes, feature extraction and classification. The aim is to come up with a simple algorithm which is computationally less intensive, while automatically generating a

classifier from the training set, using the fuzzy-rough set approach.

Figure 4.1 shows an overview of the components of the proposed recognition system. The available data is divided into training, validation, and testing datasets. The classification rules are generated in the training phase, after extracting the features of the image. The discriminative features of the images are identified and selected in the feature selection phase. In the testing phase the images are classified using the generated classification rules.

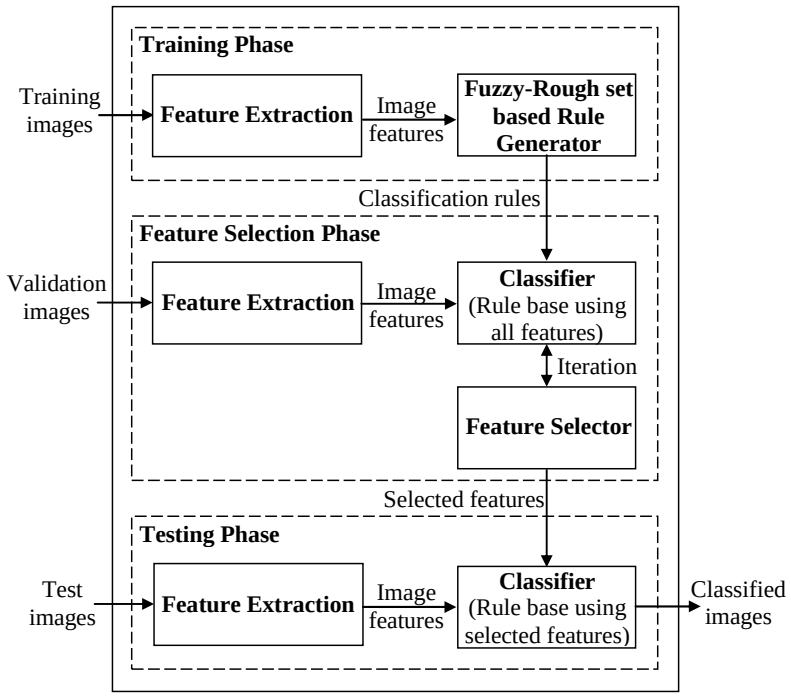


Figure 4.1: Overview of the recognition algorithm.

4.2.1 Training Phase: Identification of Feature Cluster Centers and Generation of Classifier Rules

This section explains the various steps associated with the training of the classifier. Figure 4.2 shows the training phase of the classifier. The features of the image are extracted using the C_2 feature extraction system explained in Section 2.3.1. The fuzzy discretization of the feature space is done using the cluster centers identified by the subtractive clustering technique [12]. Then the lower and upper approximations of the fuzzy sets are found out and the classification rules are generated. The following subsections explain each of these phases.

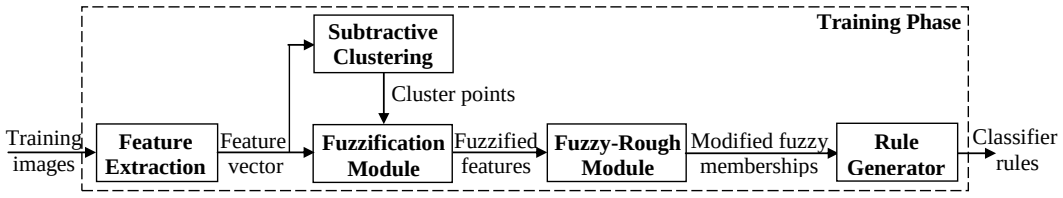


Figure 4.2: Training phase of the recognition algorithm.

Fuzzy Membership Functions from Feature Cluster Centers

The fuzzy membership functions are created in this step using the feature cluster centers, and these are utilized to partition the data into fuzzy equivalence classes. The subtractive clustering technique [12], which is an extension of Yager's mountain clustering method [112], is utilized to identify the feature cluster center points.

The cluster centers are determined for each feature (in the feature vector corresponding to the training images) of each class. In subtractive clustering, the number of clusters increases with smaller radius leading to increased computational requirements. The clustering radius used in the present work is 0.5. The number of cluster centers generated varied from 1 to 3. Once the data clusters are identified, the peaks of the triangular membership functions are placed at these points and, its left / right sides span to adjacent left / right data cluster centers. The minimum / maximum feature values position the left / right sides of the leftmost / rightmost membership functions. For example, if a set of cluster points $cp_1 = \{0.35, 0.5, 0.8\}$ are obtained for a feature A, which is within the range $[0.1, 0.9]$ for class C, then the triangular membership functions for A, for class C are as shown in Fig. 4.3.

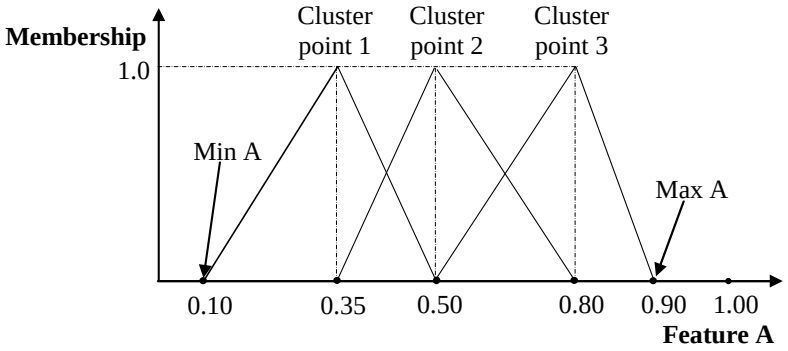


Figure 4.3: Formation of membership functions from cluster center points.

A set of membership functions is obtained for each feature of every class using the cluster center points. The training data is then fuzzified using the generated membership functions and the lower and upper approximations are obtained.

Modified Fuzzy Membership Functions

It is found that the same fuzzy equivalence class contains the samples from different output classes which causes the fuzzy-rough uncertainty. The proposed algorithm finds the lower and upper approximations of the fuzzy equivalence classes. The limiting values of the memberships that partition the definite and possible members of the output class are identified. The classification rules are generated utilizing these membership values. The limiting values of the memberships are calculated as follows (Fig. 4.4).

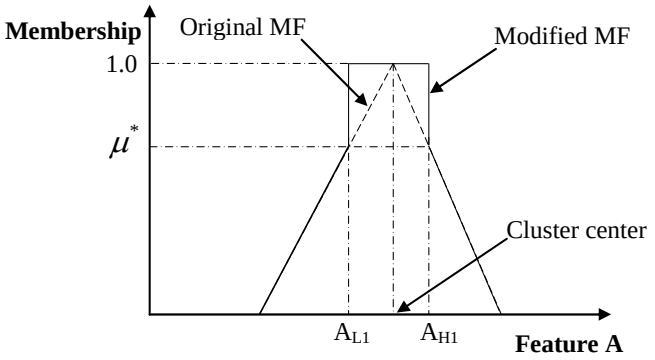


Figure 4.4: Modified fuzzy membership function.

Let MF be the fuzzy set associated with a particular cluster center point of a feature in a class and,

$$\mu^{max}(C_i) = \max\{\mu_{MF}[x_{C_i}(l)]\}, \tag{4.1}$$

where,

$x_{C_i}(l)$ is a sample from the class C_i,

and,

$$C_{max} = \text{argmax}_{C_i}\{\mu^{max}(C_i)\}, \tag{4.2}$$

then,

$$\mu^* = \max_{C_i \neq C_{max}} \{\mu^{max}(C_i)\}. \tag{4.3}$$

μ^* is the maximum of the membership values associated with data samples belonging to classes other than C_{max}. This limiting value of membership entails the rule: if any $\mu_{MF}(x) > \mu^*$ then the associated class is C_{max}. This rule implies that if the value of the particular feature A is within $[A_{L1}, A_{H1}]$ then the sample belongs to class C_{max}. These membership limits decide whether a particular sample is a definite or possible member of the output class (C_{max}). The rule always holds true for the training samples. However some of the rules classify only a small number of training samples (say one or two), if the samples from various classes are well mixed. To increase the reliability of the classifier by avoiding the generation of rules from outlying data points, only those rules which classify two or more number of training samples are stored in the rule base. In order to classify a new sample, a generalization of the rule is needed, which is provided in Section 4.2.1.

Generation of Classification Rules

The generalized classification rules are discussed in this section. Let $\{\mu^*_{ijk}\}$ be the set of membership values obtained as per Eq. (4.3) where,

i	1,. . . ,p	: p -	number of classes,
j	1,. . . ,q	: q -	number of features,
k	1,. . . ,r	: r -	number of clusters corresponding to the j^{th} feature of the i^{th} class,

then the following two rules are utilized for classifying the patterns. *Rule 1* is a voting step whereas *Rule 2* is the decision making step.

Rule 1:

$$IF\ [\mu_{MF_{ijk}}(x) > \mu^*_{ijk}]\ THEN\ [N_{C_i} = N_{C_i} + 1] \tag{4.4}$$

where,

x the sample to be classified.

N_{C_i} the number of votes for a particular class C_i,

Rule 2:

$$C = argmax_{C_i}\{N_{C_i}\} \tag{4.5}$$

where,

C is the class to which the sample belongs to.

For example in case of a dataset with three classes, the number of votes in the three classes namely N_{C_1}, N_{C_2}, and N_{C_3} are calculated for the particular sample under consideration (*Rule 1*). *Rule 2* then identifies the class with maximum votes. *Rules 1* and *2* serve to form the classifier rule base, keeping the algorithm computationally simple.

The proposed classifier is a *margin classifier* that provides the minimum distance from the classification boundary, namely *margin of classification (MC)*, for each sample. The margin of classification of a particular sample for the proposed classifier is defined as,

$$MC = N_p - N_{nMAX} \tag{4.6}$$

where,

MC	is the margin of classification,
N_p	is the number of positive votes, and
N_{nMAX}	is the maximum number of negative votes.

For a sample from class 1, let the values $N_{C_1} = 90$, $N_{C_2} = 5$, and $N_{C_3} = 10$ (in the case of a three class classification problem). Then the *MC* for the sample is $90 - 10 = 80$. In this case the sample received 90 positive votes.[1] The number of negative votes are five and ten in classes two and three respectively. A positive margin indicates correct classification whereas negative margin indicates misclassification. In Section 4.2.2, the margin of classification is used to identify the relevant features of the image.

4.2.2 Genetic Algorithm Based Feature Selection

The number of features used to describe a pattern determines the size of the search space to be explored [114]. An abundance of features increases the size of the search space, thereby increasing the time needed for classification. The cost of classification in any classification process is sensitive to the number of features used to construct the classifier.

The available features in a dataset can be categorized into four, relevant features, misleading features, irrelevant features, and redundant features (Section 3.1). The removal of misleading, irrelevant and redundant features reduces the computational expenses and increases the classification accuracy. This process is known as attribute reduction or, in the context of machine learning, feature selection.

[1] Voting is positive if the voted class and the actual class are the same. Otherwise it is negative.

In the present work the feature selection is done in two stages. The first stage is a pre-filter which filters out the misleading features. The second stage is a GA based feature selection algorithm which removes the irrelevant and redundant features. The feature reduction process is depicted in Fig. 4.5.

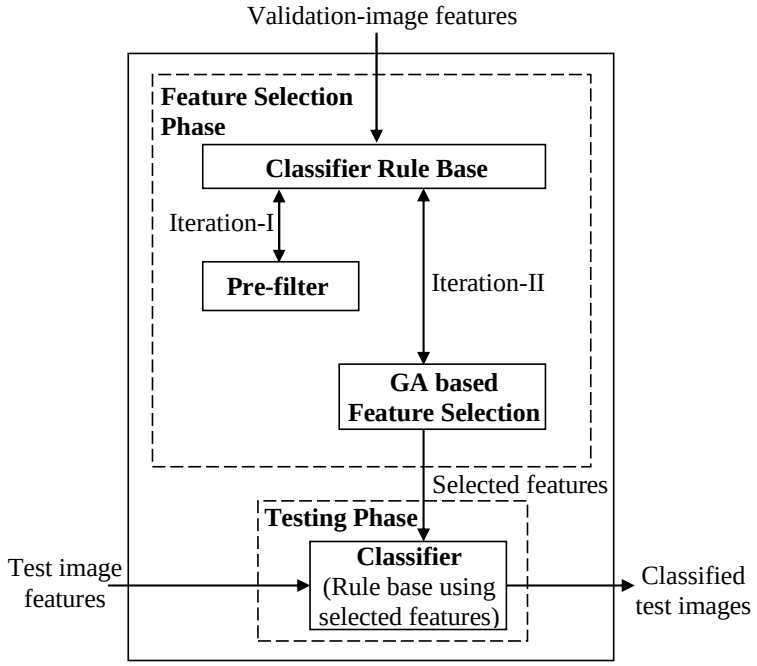


Figure 4.5: Feature selection and testing phase.

Each classification rule in the proposed classifier corresponds to a particular feature of the image. The classification rules provide positive or negative votes to the samples. This helps to identify the relevant features (the features that provide more positive votes) of the image. The proposed margin of classification (4.6) is utilized for the feature selection process. Selection of the relevant features improves the performance of the recognition system and reduces the computational burden of the overall algorithm.

Pre-filter: Weeding-out the Misleading Features

Pre-filter (Fig. 4.6) identifies and removes the features that give negative votes in the classification process. The features, which negatively vote for two or more samples as per (4.4), are weeded out. The validation data is sorted in the ascending order of the margin of classification and those samples which have less *MC* are considered (the samples which have less *MC* have more negative votes and are nearer to the classification boundary). The features which vote these samples negatively are then identified and removed from the dataset. For example if $x(1)$, $x(2)$, $x(3)$ and $x(4)$ are the first four samples in the sorted list, and, A_1, A_2, A_3 and A_4 are the subsets of features (the misleading feature subsets) which give negative votes to these samples respectively, then the set of features eliminated is A', where A' consists all the features which are members in at least any two of the subsets A_1, A_2, A_3 and A_4. The process is iterated a few times by repeating the validation procedure. The number of iterations needed depends on the nature of the dataset. More number of iterations are undergone for a dataset with large number of misleading features. In the present work the number of iterations used is three. The elimination process is a simple operation but it has significant effect in increasing the margin of classification, and in reducing the computational load at the second stage (as the number of classification rules is reduced).

Feature Selection using Genetic Algorithm

Genetic Algorithms (GAs) are widely used for feature selection and extraction in machine learning [6, 44, 55, 82, 89]. In the present work GA is

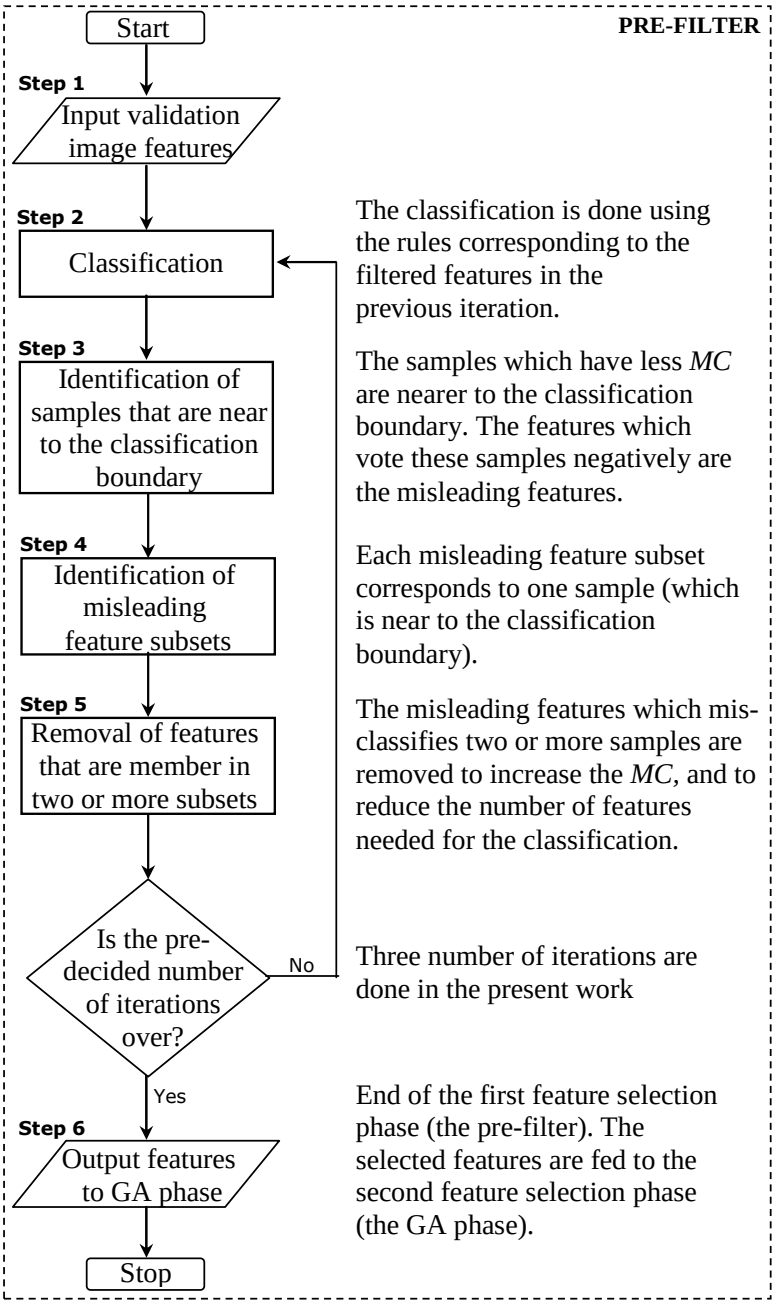


Figure 4.6: Flowchart of the pre-filter.

utilized to remove the irrelevant and redundant features. A fitness function is defined which considers the accuracy, cost, and margin of classification in accordance with the classification algorithm proposed in Section 4.2.1.

Each chromosome in the population represents the set of features used for classification. A chromosome consists of N_p bits and each bit represents a specific feature. When a particular bit is unity, it indicates the presence of the feature and its absence is indicated by a zero. In the present work the value of N_p is 200. Elitism technique is utilized to create a new population and the fitness function Eq. (4.7) is minimized.

$$F = w_1 * N_1 + w_2 * N_2 - w_3 * N_3, \tag{4.7}$$

where,

N_1	number of misclassifications,
N_2	number of features used for classification,
N_3	minimum ***MC*** (minimum of ***MC***s of different samples),
w_1, w_2, w_3	weighing factors.

N_1 represents the accuracy of classification and N_2 represents the computational expense (presence of more features leads to more classification rules and so is the computational cost). N_3 provides preference to chromosomes with better margin of classification. The minimum margin of classification is the margin of the sample which is nearest to the classification boundary. $w_1, w_2,$ and w_3 are to be tuned depending on the dataset. An empirical guideline is to select the weights (w_1, w_2, w_3) such that $w1 > w2 > w3$, which provides the first preference to the accuracy of classification, second preference to the number of features and third

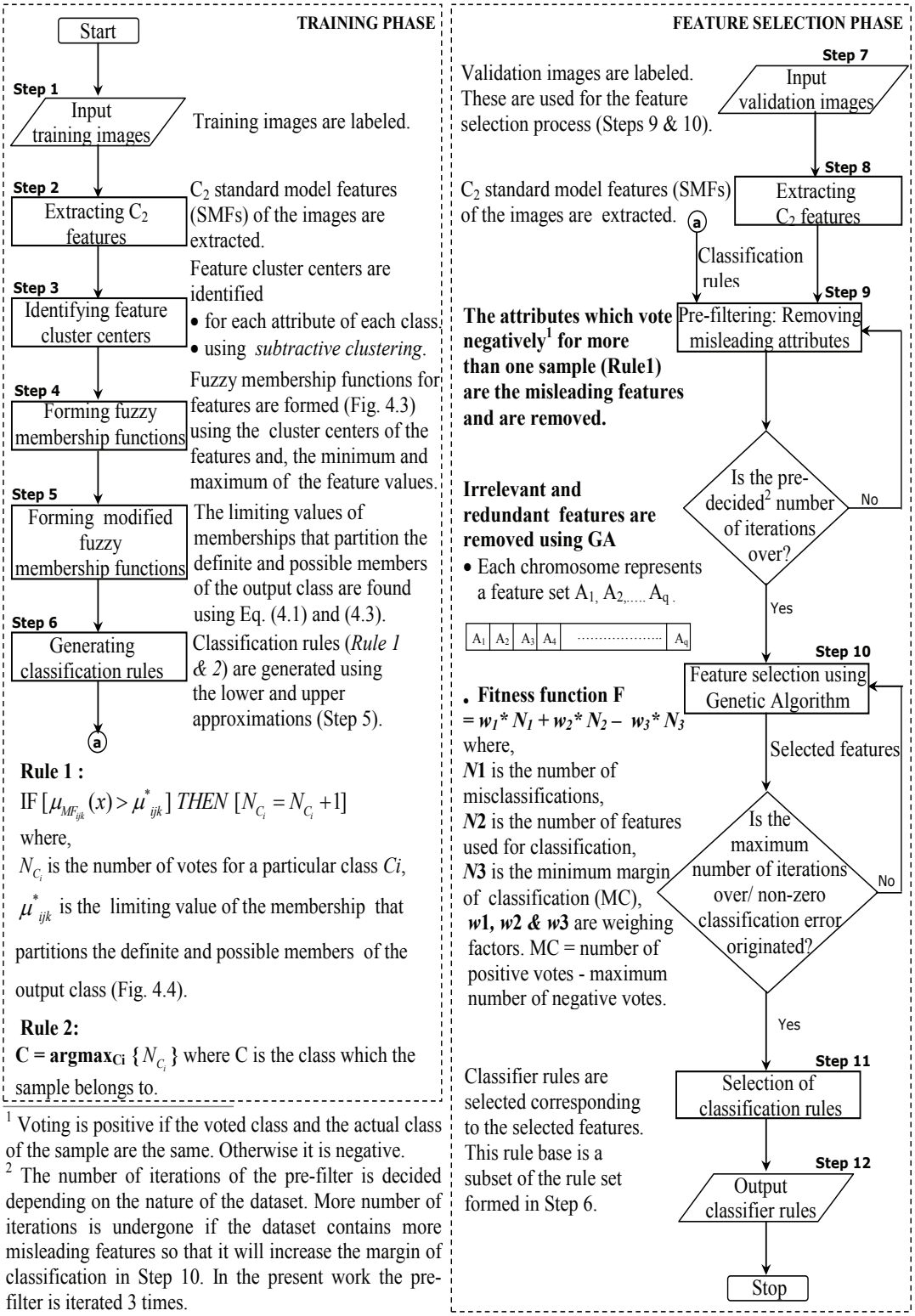
preference to the margin of classification. In the present work the values of $w_1, w_2,$ and w_3 are $1.0, 0.6,$ and 0.4 respectively.

The crossover and mutation rate used are 0.9 and 0.01 respectively. The population size of the GA is kept at 100. Iterations are done until the pre-decided maximum number of iterations (50 in the present work) is reached or till a non-zero classification error is originated. If iteration is stopped after the origination of a non-zero classification error, then the features used in the previous iteration is selected. The number of features needed for classification is decided by taking the average over ten runs of GA.

A detailed flowchart of the classifier development algorithm is provided in Fig. 4.7.

4.2.3 Testing Phase: The Classifier

Fig. 4.8 shows the flowchart of the testing phase of the recognition system. The selected features of the unlabeled test images are extracted using the C_2 feature extraction system. Then the memberships of these features in different classes are calculated using the membership functions formed in the training phase. This membership values are compared with the limiting value of the membership function (4.3) and the classification is done using *Rules 1* and *2*. Each of the execution steps of the classifier consists of a few number (which is equal to the number of cluster centers of the feature) of comparisons, using the classification rules, which makes the algorithm computationally simple. The classification results are discussed in Section 4.3.



[1] Voting is positive if the voted class and the actual class of the sample are the same. Otherwise it is negative.

[2] The number of iterations of the pre-filter is decided depending on the nature of the dataset. More number of iterations is undergone if the dataset contains more misleading features so that it will increase the margin of classification in Step 10. In the present work the pre-filter is iterated 3 times.

Figure 4.7: Flowchart of the classifier development algorithm.

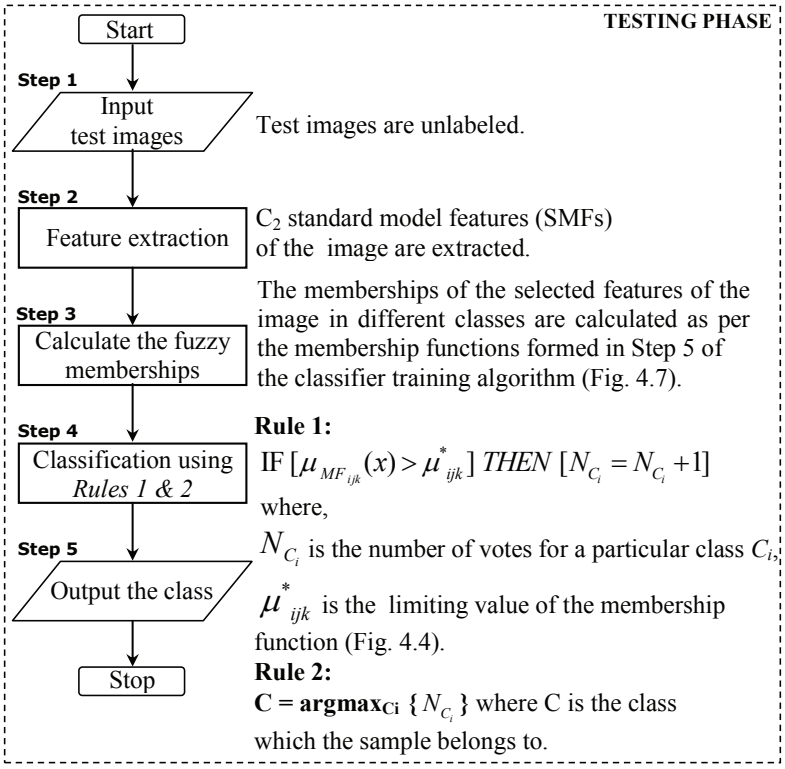


Figure 4.8: Flowchart of the testing phase.

4.2.4 Computational Complexity Analysis

The different parameters at the input of the recognition system are the number of classes and the size of the input image. The complexity of the recognition algorithm has two components, complexity due to the feature extraction system and that due to the proposed classifier. This section discusses the computational complexity of the classifier algorithm.

The fuzzy-rough pattern classifier is formed using the classification rules generated in the training phase. The pseudo code of the classifier is shown in Fig. 4.9. Let p be the number of classes and q be the number of selected features at the input of the classifier. The complexity of the algorithm is as follows : $O(q)$ for reading the image features, $O(pq)$

for reading the trained parameters of the classification rules, $O(pq)$ for the fuzzy membership calculation and voting process, and, $O(p)$ for finding the class index which received the maximum number of votes. So the overall complexity of the algorithm is polynomial time, $O(pq)$. As the feature extraction algorithm provides a fixed number of features, independent of the input image size, the effective complexity of the classification algorithm reduces to $O(p)$.

```
read {test_image_features} ;  % read the selected C2 features.
read { μ*}, {Amin}, {Amax}, {Acenter} ; % read the membership limits, min. & max.
                                        % values of features, and feature cluster centers.
for i =  number_of_classes
for j = number_of_features;  % number of selected features.
     calculate {fuzzy_memberships} ; % calculation of  μ .
          IF  μ > μ*
               vote(i) = vote(i) +1;
          end
     end
end
class =  index {max (vote)} ; % find the class index which received maximum votes.
print {class} ;
```

Figure 4.9: Pseudo code of the classifier.

4.3 Experimental Evaluation

The proposed algorithm is tested using three face datasets and two hand posture datasets (Table 4.1). The datasets contain ten classes with equal number of samples. The dataset is divided into N subsets, with equal

number of images. The images in the first subset are used for the development of the classifier while those in the remaining $N-1$ subsets are used for testing. The classifier development is then done using the second subset and the algorithm is tested using the remaining $N-1$ subsets. The experiments are repeated in a similar fashion, N times, until each of the subset is used for the development, and, the average accuracy achieved over the N runs is reported. The values of N are two and four ($N = 2, 4$ corresponds to $50\%, 25\%$ images in one subset respectively). The data subset used for the classifier development is equally divided into training and validation sets. The classification results are compared with that of Support Vector Machines (SVM) with polynomial kernel, implemented using LIBSVM [7].

Table 4.1: Details of face and hand posture datasets

Dataset	**Source**	**# images**
Face dataset-1	Subset of Yale face database B [23]	640
Face dataset-2	Subset of color FERET database [69]	240
Face dataset-3	Subset of CMU face dataset [90]	240
Hand posture dataset-1	NUS hand posture dataset	240
Hand posture dataset-2	Jochen Triesch hand posture dataset [95]	480

4.3.1 Face Recognition

The proposed recognition algorithm is tested using three different face datasets which have variation in lighting direction (Yale face database B [23]), variation in pose (color FERET database [69]), and illumination variation (CMU face dataset [90]).

The first face dataset considered is a subset of the Yale face database B [23], which contains ten classes of face images, taken from different

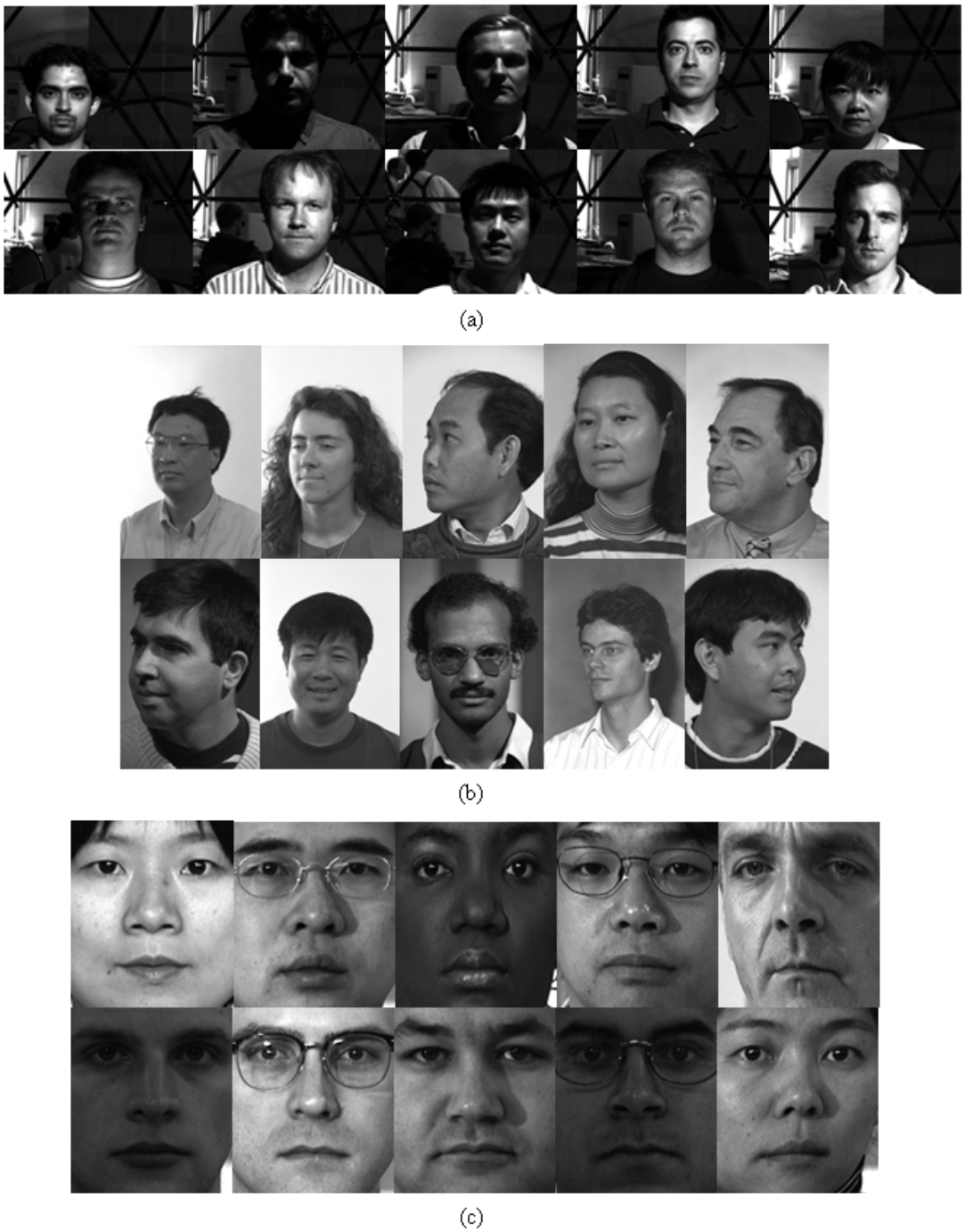


Figure 4.10: Sample images from (a) Yale face dataset, (b) FERET face dataset, and (c) CMU face dataset.

lighting directions (Fig. 4.10(a)). It consists of 640 frontal face images. A comparison of the achieved results with that of SVM using C_2 features is provided in Table 4.2. The fuzzy-rough classifier provided full classification of the dataset whereas SVM provided 99.38% accuracy, when $N = 2$. The proposed algorithm classified the dataset utilizing only 86 number of features, whereas SVM utilized all the 1000 features.

The second face dataset used is a subset of the color FERET database [69]. The subset contains ten classes of face images, with variations in pose (Fig. 4.10(b)). It consists of 240 images. A comparison of the results with that of SVM is provided in Table 4.2. The proposed algorithm needed only 102 features (when N=2) for the classification and it provided better accuracy compared to SVM classifier.

Table 4.2: Recognition results - face datasets

Dataset	**No. of training samples**	**FRC***		**SVM**	
		Accuracy (%)	No. of features	Accuracy (%)	No. of features
A subset of Yale face dataset [23]	160 (N=4)	99.16	95	98.33	1000
	320 (N=2)	100	86	99.38	
A subset of color FERET database [69]	60 (N=4)	88.89	111	88.33	
	120 (N=2)	91.66	102	90.83	
A subset of CMU face dataset [90]	60 (N=4)	99.44	62	98.88	
	120 (N=2)	100	55	100	

*Fuzzy-Rough Classifier

The third face dataset considered is a subset of the CMU face dataset [90]. It has good amount of illumination variation as shown in Fig. 4.10(c). The dataset consists of 240 frontal face images (24 from each class). The recognition results are provided in Table 4.2. Both the proposed algorithm and the SVM classifier classified the dataset fully. However the proposed

algorithm did the classification with lesser number of features.

4.3.2 Hand Posture Recognition

As the number of available hand posture datasets is limited, a new dataset namely the NUS hand posture dataset,[2] with ten classes of hand postures, is created. It consists of 24 sample images per posture, which are captured by varying the position and size of the hand within the image frame. These are gray-scale images (160 x 120 pixels). The hand postures are selected in such a way that the inter class variation in the appearance of the postures is less, which makes the recognition task more challenging. Sample images are shown in Fig. 4.11(a). The better performance of the proposed algorithm is evident from the recognition results (Table 4.3). The proposed algorithm utilized only 92 number of features (when N=2) for the classification. The misclassifications occurred are mostly between the classes which are similar.

The second hand posture dataset considered is Jochen Triesch hand posture dataset [95]. The ten classes of hand postures (Fig. 4.11(b)), performed with 24 different persons, against light and dark backgrounds, were considered for the classification (total of 480 images). The images vary in size of the hand and shape of the postures. The proposed algorithm achieved better classification accuracy on comparison with SVM (Table 4.3). The classification is done utilizing 63 selected features (when N=2), whereas the SVM classifier utilized all 1000 features.

[2]This dataset is available on an e-mail request to prahlad@nus.edu.sg

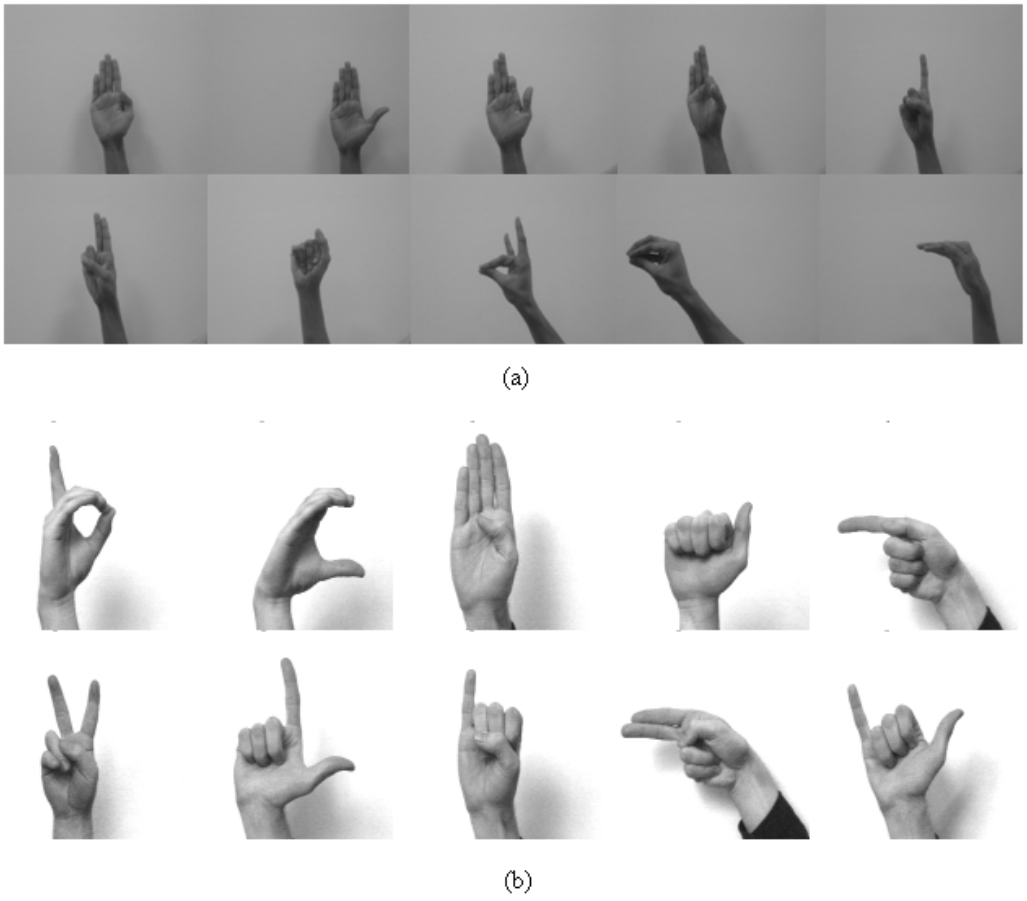


Figure 4.11: Sample hand posture images from (a) NUS dataset, and (b) Jochen Triesch dataset.

Table 4.3: Recognition results - hand posture datasets

Dataset	**No. of training samples**	**FRC***		**SVM**	
		Accuracy (%)	No. of features	Accuracy (%)	No. of features
NUS hand posture dataset	60 (N=4)	91.66	98	91.80	1000
	120 (N=2)	93.33	92	92.50	
Jochen Triesch hand posture dataset [98]	120 (N=4)	95.83	72	94.44	
	240 (N=2)	98.75	63	97.91	

*Fuzzy-Rough Classifier

4.3.3 Online Implementation and Discussion

The proposed recognition system is implemented online in windows platform to test the computational performance. The algorithm is developed using the NUS hand posture dataset for the recognition of hand postures. The image to be recognized is accessed using a web-camera, resized and converted to gray scale, similar to the training images.

The algorithm is tested by showing each hand posture 20 times. The hand postures are performed by different persons, with variation in size and shape of the hand posture, and with different lighting conditions. The algorithm recognized the postures with an accuracy of 94.5%.

Table 4.4: Comparison of computational time

	Proposed algorithm	**SVM**
Feature extraction	1.92 s	7.43 s
Classification	0.96 ms	3.24 ms

Table 4.4 provides a comparison of the average computational time of the proposed algorithm with that of SVM. The total time for recognition is divided into two, the time for the feature extraction, and the time for the classification. In the SVM classifier all the 1000 features are utilized for the classification, and so 1000 features are extracted. The proposed algorithm needs only a subset of the 1000 features. This reduced both feature extraction and classification time (The proposed classifier took lesser time for classification even when 1000 features are utilized).

Each of the extracted C_2 SMF corresponds to a particular prototype

patch with a specific patch size, as explained in Section 2.3.1. The selection of relevant features identified the prototype patches those are good in the inter class discrimination. This enhanced the shape selectivity. Also it decreased the processing time for the feature extraction, which is a major limitation of the feature extraction algorithm proposed by Serre *et al.* [85]. The significant reduction in the feature extraction and classification time makes the algorithm suitable for real-time applications.

4.4 Summary

A hand posture and face recognition algorithm using C_2 standard model features (SMFs), and based on the concepts of lower and upper approximations of fuzzy equivalence classes, is proposed in this chapter. The fuzzy membership functions and the corresponding classification rules are generated from the training images and the classification is done by a simple voting process. The predictive features in the dataset are selected using a GA based feature selection algorithm. The proposed fitness function reduces the number of features required for the classification, without compromising on the classification accuracy. The performance of the algorithm is evaluated with some well-known datasets. The recognition results are compared with that of an SVM classifier. The proposed algorithm provided good recognition accuracy for all the datasets considered, at a less computational cost, which makes it suitable for real-time applications.

Chapter 5

Hand Posture Recognition using Neuro-biologically Inspired Features

5.1 Introduction

Chapters 3 and 4 presented two feature selection and classification algorithms for pattern classification. In these algorithms the image features are extracted using a standard model of the visual cortex proposed by Serre *et al.* [85]. The visual object recognition is mediated by the ventral visual object processing stream in the visual cortex [26, 81]. Serre *et al.* [85, 86] proposed a computational model of the ventral stream, based on the standard model of visual object recognition [81]. The model provided a hierarchical system which comprises four layers that imitates the feed-forward path of object recognition in the ventral stream of primate visual cortex (Section 2.3.1). This chapter suggests modifications and improvements in the system, for the extraction of hand posture features.

Two different algorithms, based on the computational model of the ventral visual object processing stream, are proposed for hand posture recognition.

5.2 Graph Matching based Hand Posture Recognition using C1 Features

Graph matching plays a key role in many areas of computing from computer vision to networking, where there is a need to determine correspondences between the components of two attributed structures. Graph matching is a graph representation and matching technique and is a powerful tool in computer vision due to its high representational nature and the ability to handle complex images.

In a typical graph representation, regions of the image are represented by nodes in a graph. These nodes are related to each other by edges, which express the structural relationships between regions. The nodes of a graph are labeled with a local image description, and the edges are labeled with a distance vector.

Triesch *et al.* employed elastic graph matching (EGM) technique to develop a person independent hand posture recognition system [97, 98]. EGM is a neurally inspired object recognition technique [47], which has wide applications in general object recognition, face finding and recognition, gesture recognition, and the analysis of cluttered scenes. In EGM the model graphs (the graphs generated from the training images) are

matched with the input image which contains the pattern to be recognized. Elastic matching of a model graph to an image means to search for a set of node positions such that 1) the local image description attached to each node matches the image region around the position where the node is assigned and 2) the graph is not distorted too much [97].

The present work utilizes a new set of image features, extracted using the computational model of the ventral stream of visual cortex [85, 86], to label the various nodes in the graph. The model graph is created by assigning the nodes to geometrically significant positions in the hand image. The classification of the hand posture is done by identifying the best match between the input image and the model graphs. A radial basis function (RBF) is used as the similarity function for the comparison of node features. The proposed algorithm is tested on a 10 class hand posture database. It provided better recognition accuracy compared to the reported results in the literature.

5.2.1 The Graph Matching Based Algorithm

The proposed algorithm utilizes the structure of distribution of the graph nodes to understand and recognize the shape of the hand posture. Each node is labeled using Gabor filter based features (the C_1 features). Gabor filters are edge detecting filters. The responses of the image to Gabor filters with various orientations store the shape information of the hand. In the presented algorithm, the node features are expressed using a C_1 image patch, which contains the local shape information around the node. A group of such nodes stores the complex shape of the hand posture.

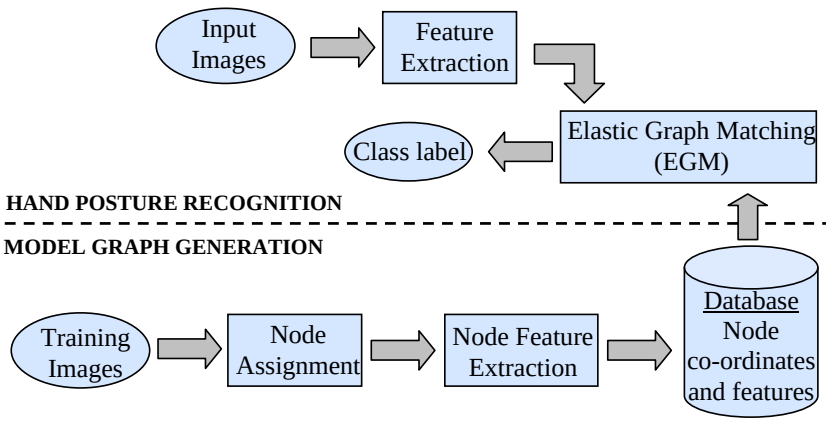


Figure 5.1: The graph matching based hand posture recognition algorithm.

Fig. 5.1 shows a simplified block diagram of the overall algorithm. The model graphs are created from the C_1 response of the training images. The nodes of the model graphs are assigned to the geometrically significant positions in the hand image (Fig. 5.2(a)). The C_1 images (Fig. 5.2(b)) of the training images are extracted and these are sub-sampled (1 pixel per 5 pixels - to reduce the computational burden). The sub-sampled C_1 image patches centered at the graph nodes are extracted. The patch size used is 15 x 15 pixels. Each patch contains four orientations $(0°, 45°, 90°, -45°)$. These image patches are used as the model graph node features.

In [85] RBFs are used in the S_2 stage which compares the complex features at the output of C_1 stage (which corresponds to the retinal image) with patches of previously seen visual image and shape features. RBFs are a major class of neural network model, which compare the distance between input and a prototype [5]. The proposed algorithm utilizes RBF

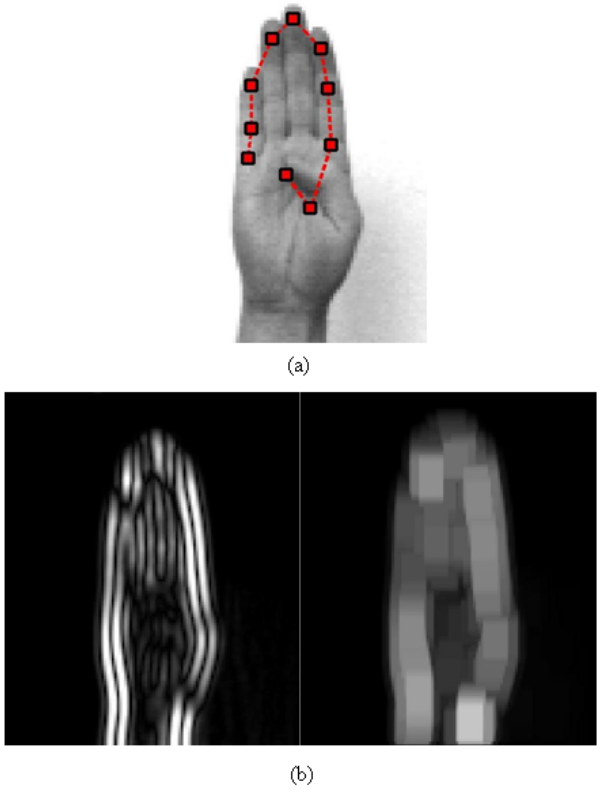


Figure 5.2: (a) Positions of graph nodes in a sample hand image, (b) S_1 and C_1 responses of the sample image (orientation $90°$).

as the similarity function for matching the graphs. The similarity of each node depends in a Gaussian-like way on the Euclidean distance between the node feature of the input image and that of the model graph. The smaller the Euclidean distance, the bigger will be the similarity.

The algorithm utilizes several images of the same class (four images per class) to create the model graph, as that done in bunch graph approach [107]. Bunch graphs model the variability in object appearance. In a bunch graph, each model graph node is represented using a bunch of node features. The bunch of features is extracted from identical nodes (the nodes assigned to identical positions) of several images, which vary in size and shape of the posture. During the recognition process, all the features from the bunch is matched, and the maximum of the similarities

is considered for the decision making.

For the recognition of a hand posture, the model graphs are sequentially matched to the input image as described below.

1. The C_1 image of the input image is extracted.
2. The C_1 image is sub-sampled (1 pixel per 5 pixels).
3. The sub-sampled image is scanned in x and y direction using the model graphs.
4. The similarity between each graph node i is calculated using the equation

$$s_{ij} = \exp(-||X_i - P_{ij}||^2), \tag{5.1}$$

   where X_i is the feature (C_1 image patch) of the i^{th} input image node and P_{ij} is the feature[2] of the i^{th} node of j^{th} class (model graph). Each node is allowed to shift its position by two pixels in the x and y direction (two pixels left, right, above and below). This is to take care small distortions in the graph.
5. The similarity S_j between the two graphs is calculated by summing up the node similarities (5.2).

$$S_j = \sum_{i=1}^{n} s_{ij}, \tag{5.2}$$

   where n is the number of nodes in the model graph.
6. The class corresponding to the model graph which provides the best match is declared as the output class C, the class to which the posture belongs to (5.5).

$$C = \text{argmax}_j\{S_j\} \tag{5.3}$$

Fig. 5.3 shows a detailed flowchart of the model graph generation and the hand posture recognition algorithm.

[2]The feature from the bunch of node features, which provides the maximum similarity.

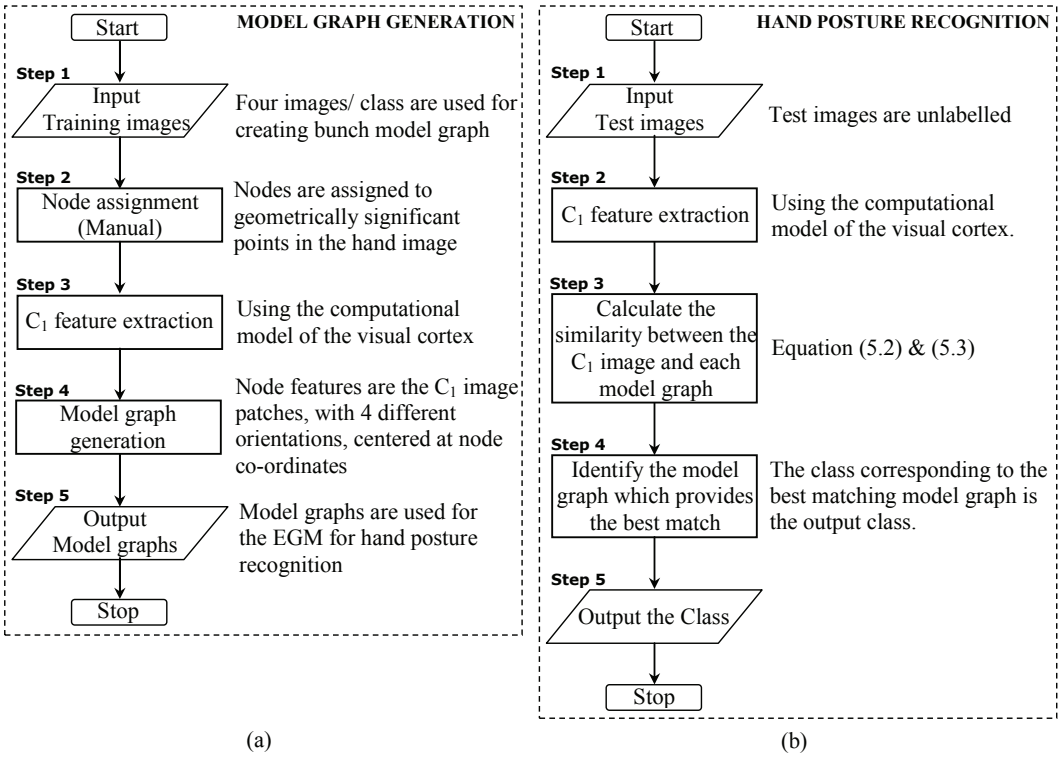


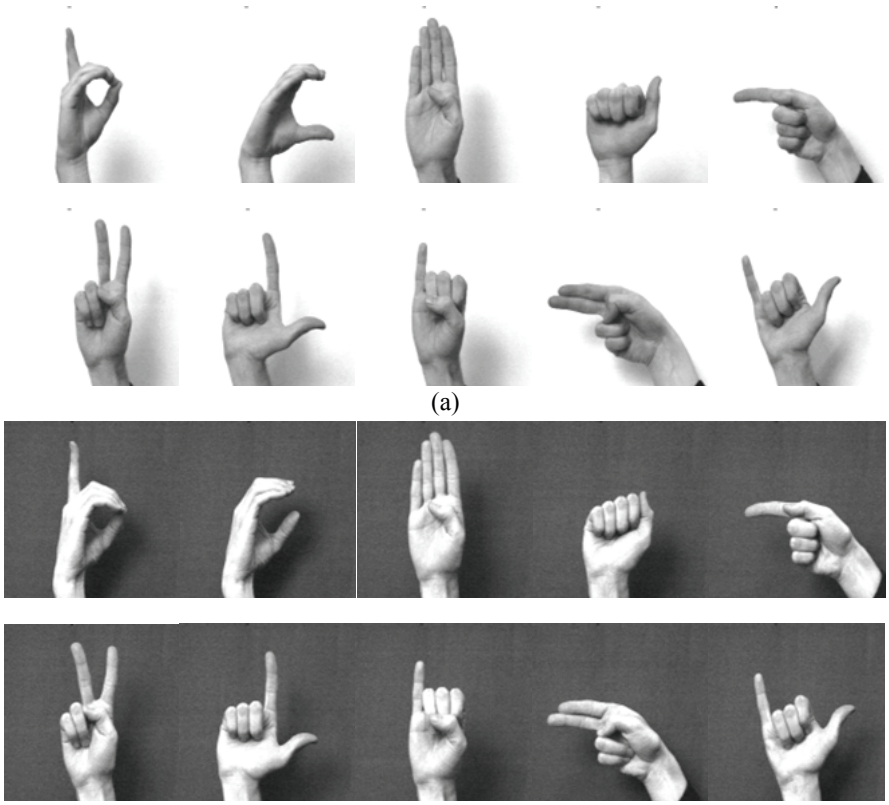
Figure 5.3: Flowchart of (a) the model graph generation and (b) the hand posture recognition algorithm.

5.2.2 Experimental Results

The proposed algorithm is tested on Jochen Triesch hand posture dataset [95]. The dataset consists of 10 classes of hand postures, performed by 24 persons, against light and dark backgrounds (Fig. 6.9). The images are 8-bit grey-scale images of 128 x 128 pixels. A total of 240 light background images and 238 (2 images missed) dark background images are provided. The images vary in size and shape of the hand posture.

The hand postures of two persons against light and dark backgrounds (40 images) are used for the generation of model graphs. The algorithm is tested using the remaining images (438 images). Out of 220 test images with light background (which are not used for model graph generation)

213 were recognized correctly, which corresponds to a recognition accuracy of 96.82%. Recognition accuracy for dark background images is 209 out of 218, which is 95.87%. Most of the errors occurred when the algorithm failed to discriminate between postures 5 and 9 (Fig. 6.9) which look similar. The average accuracy provided by the algorithm is 96.35% (Table 6.6), whereas the earlier reported accuracy [98] for the same dataset is 93.77%. In [98], 60 images (6 images per class), each with 35 nodes, are used for the model graph generation, whereas in the proposed algorithm the model graphs are generated using 40 images (4 images per class), each with 10 nodes. The reduction in number of nodes reduces the computational burden of the recognition algorithm.



(a)

(b)

Figure 5.4: Sample hand posture images (a) with light background and (b) with dark background, from Jochen Triesch hand posture dataset [95].

Table 5.1: Comparison of recognition accuracy

Background	# test images	# correct classifications	Proposed method	Reported in [98]
Light	220	213	96.82	94.3
Dark	218	209	95.87	93.3
Net	438	422	**96.35**	93.77

The total computational time needed for the recognition process is divided into two, the time for the feature extraction, and the time for the classification. The feature extraction takes 1.9 seconds (on an average) to complete, which needs to be improved for real-time applications. The classifier, which is based on elastic graph matching, has a high computational complexity. However the present work utilized only 10 number of nodes per model graph, which limited the classification time to few milliseconds (12 ms on an average, on a computer with 3 GHz Intel Pentium 4 CPU).

5.2.3 Summary

An EGM algorithm using a new set of biologically inspired features is proposed for hand posture recognition. The nodes of the graph are labeled using the C_1 SMFs, which are extracted using the computational model of the ventral stream of visual cortex. The shape of the hand posture is recognized by identifying the structure of distribution of nodes in the input image. The graphs are matched using a radial basis function as the similarity function. The variations in the size and shape of the hand posture is taken care by the local maximization in the C_1 stage, and by using the bunch graph approach. The algorithm is tested on a 10 class hand posture dataset. It provided better recognition accuracy compared

to the earlier reported results. The algorithm is scale tolerant and person independent.

5.3 C2 Feature Extraction and Selection for Hand Posture Recognition

The computational model proposed by Serre *et al.* extracts the scale and position tolerant C_2 Standard Model Features (SMFs). A major limitation of this model in real-world applications is its processing speed. Also the algorithm needs a separate classifier for the classification of the pattern. This section proposes a modification of the above algorithm which does not need a separate classifier stage, and which is computationally efficient. The prototype C_1 patches used for learning are selected from specific classes. The patches which have good interclass differences are identified and the discriminative C_2 features are extracted. The classification is done by a comparison of the extracted C_2 features.

5.3.1 Feature Extraction and Selection

Serre *et al.* suggested that it is possible to perform robust object recognition with C_2 SMFs learned from a separate set of randomly selected natural images. However it was agreed that *object specific* features (the features learned from images which contain the target object) perform better compared to the features learned from random images. The proposed algorithm extracts features by considering the class division. The modification is done at the third layer of the C_2 feature extraction system.

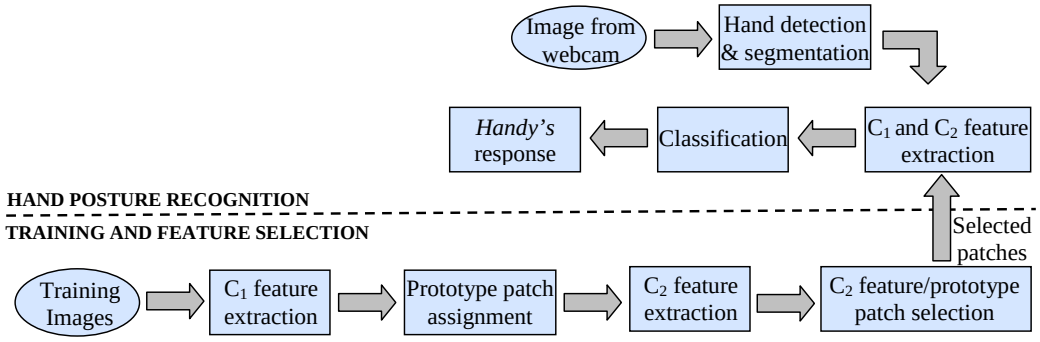


Figure 5.5: The C_2 features based hand posture recognition algorithm.

Instead of selecting the patches from random positions and random images, they are selected from geometrically significant positions of specific hand posture classes. In [85] support vector machines (SVM) and boosting based classifiers are used for the classification, whereas the proposed algorithm does the classification by a simple comparison of the extracted features. Also the patches which have more inter-class differences are identified and selected which increased the classification accuracy and reduced the computational cost (by reducing the number of features). Fig. 5.5 shows a block diagram of the proposed algorithm. The different blocks of the diagram are explained in the following subsections.

Assigning Prototype Patches Specific to Posture Classes

The C_1 features of the training images (one image/class) are extracted. The prototype patches (center of the RBF units) of different sizes are assigned to the C_1 images of specific classes. N(= 10) patches, each with Q(=4) different patch sizes are extracted from M(= 10) classes. The patches are located at geometrically significant positions in the hand image (Fig.

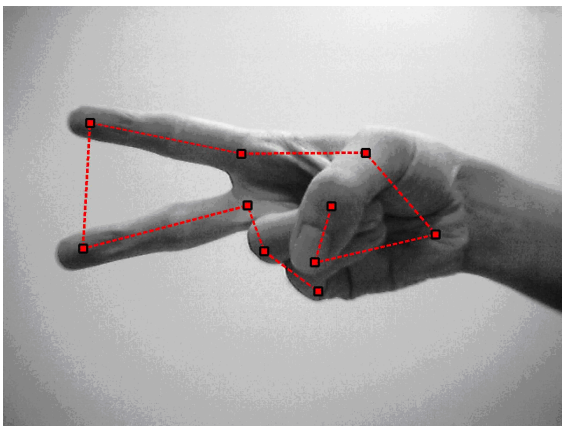


Figure 5.6: Positions of prototype patches in a sample hand image.

5.6). Each patch contains C_1 responses with O(= 4) different orientations. The C_2 features are extracted from the C_1 images using the prototype patches. Each of the extracted C_2 feature corresponds to a particular patch with a specific size and totally there are QxMxN C_2 features.

Classification

Once the C_2 features are extracted the classification is done as follows. Let

$$P_{ij} = \sum_{k=1}^{Q} C_2(ijk), \tag{5.4}$$

where $C_2(ijk)$ is the C_2 feature corresponding to the j^{th} patch from the i^{th} class with k^{th} patch size, then,

$$C = \text{argmax}_{C_i}\{Q_{C_i}\}, \tag{5.5}$$

where,

$$Q_{C_i} = \sum_{j=1}^{N} P_{ij}, \tag{5.6}$$

and C is the class to which the hand posture belongs to.

Selection of the Discriminative C_2 Features

The C_2 features are selected based on the interclass differences in the C_2 features of the training images. Each C_2 feature corresponds to a particular prototype patch. The prototype patches which helps to distinguish different classes are identified and selected to make the extracted C_2 features discriminative. Let

$$P'_{ij} = \max_{C_l \neq C} \{P_{ij}(C_l)\}, \tag{5.7}$$

where $P_{ij}(C_l)$ is the response P_{ij} (5.4) of an image from class C_l, then $P_{ij} - P'_{ij}$ represents a *margin of classification(MC)*. The prototype patches from each class are sorted according to the descending order of the corresponding *MCs* and the first *N'*(= 5 in this paper) patches are selected.

The features of the test image are extracted using these selected discriminative patches to get the discriminative C_2 features. The hand posture classification is done using (5.5) and (5.6) (*N*=*N'* in 5.6).

5.3.2 Real-time Implementation and Experimental Results

The proposed algorithm is implemented in real-time in windows XP platform, for the interaction between human and a virtual character *Handy*. The programming is done using C# language. The virtual character is

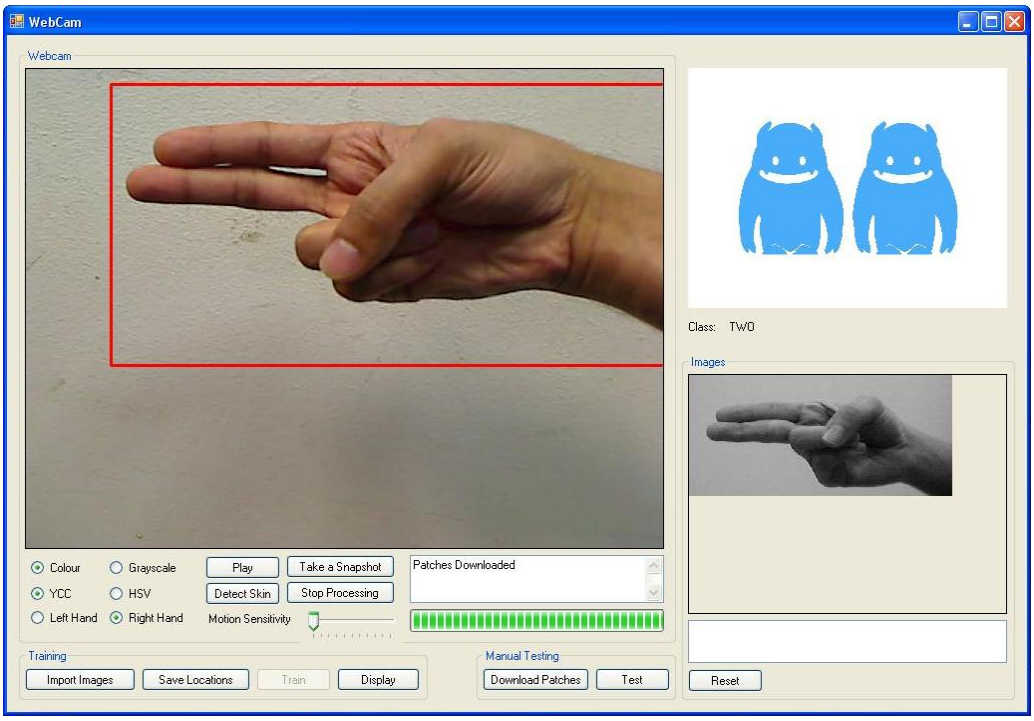


Figure 5.7: The user interface.

created using Adobe flash. Twelve different animations (10 to represent different predefined classes, 1 to represent non-class, and 1 to represent idle mode) are developed. A text to speech converter is used to generate voice response. *Handy* welcomes the user and gives the command to connect the camera, if it is not connected already. Once the camera is connected and the posture is shown, *Handy* identifies it and responds to the human by symbolically showing the posture. Also it pronounces the class number. If the posture does not belong to any of the trained classes *Handy* shows exclamation and asks *what is that?*.

The RGB image is captured using a webcam. The image is converted

to YCbCr color space and the hand is detected using skin color segmentation.[1] The segmented image is converted to gray scale and used for further processing. The hand image is captured only if the movement of the hand is less than a threshold value, which is adjustable. Low threshold value provides high motion sensitivity (more motion sensitivity means the hand posture will be detected only if the hand motion is lesser). The hand motion is detected by subtracting the subsequent frames in the video. A graphical user interface (Fig. 5.7) is developed which has provisions for displaying the skin colored area, left / right hand selection, varying the motion sensitivity of the hand, and, the training and testing of the recognition algorithm. The user interface displays the input video (on left). The posture class number, the corresponding hand posture image from the library, and the response of *Handy* are also displayed (on right) in the user interface.

The algorithm is trained for 10 classes of hand postures (Fig. 5.8). Only one image per class is used for the training and the testing is done by showing each posture 20 times. The hand postures are performed by different persons, with large variation in size and shape of the hand posture, and with different lighting conditions. The algorithm recognized the postures with an accuracy of 97.5%. The average recognition time taken is few milliseconds.

[1]The segmentation is also done in HSV color space. On comparison, segmentation in YCbCr color space provided better results.

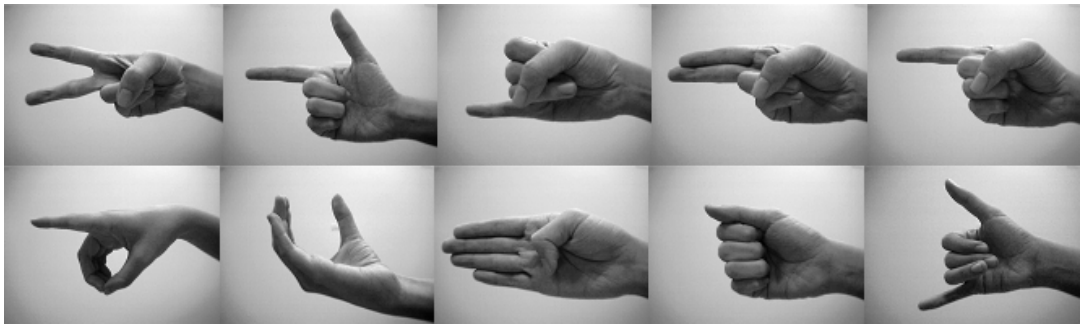


Figure 5.8: Hand posture classes used in the experiments.

5.3.3 Summary

A computationally efficient and reliable hand posture recognition algorithm is proposed for the interaction between human and computers. The proposed algorithm is motivated by the computational model of the ventral stream of visual cortex. The C_2 SMFs of the hand images are extracted in such a way that it provide good discrimination between different classes, and the classification is done by a comparison. The algorithm is implemented in real-time for the interaction between human and a virtual character *Handy*. The algorithm recognized the hand postures at a faster speed, with reliable accuracy. The recognition accuracy of the proposed classifier shows its good discriminative power. Also the proposed system has person independent performance.

The proposed algorithm easily recognizes hand postures with light and dark backgrounds. However the performance of the algorithm in complex backgrounds is to be improved. A person independent hand posture recognition system against complex natural backgrounds is presented in Chapter 6.

Chapter 6

Attention Based Detection and Recognition of Hand Postures Against Complex Natural Backgrounds

The mainstream computer vision research has always been challenged by human vision, and the mechanism of human visual system is yet to be understood well, which is a challenge for both neuroscience and computer vision. Intermediate and higher visual processes in primates select a subset of the available sensory information before further processing [101], in order to reduce the complexity of scene analysis. This selection is implemented in the form of a *focus of attention* [60]. Recent developments in the use of neurobiological models in computer vision tries to bridge the gap between neuroscience, computer vision and pattern recognition [11, 32, 33, 72, 80, 81, 85, 88].

Eng-Jon *et al.* proposed an efficient algorithm for hand detection and

static hand shape recognition [19]. The algorithm provided good detection and recognition accuracy. A robust hand posture detection algorithm based on the object recognition method proposed by Viola and Jones is presented in [45]. The algorithm is applied for the detection of six hand postures. View independent recognition of hand postures is considered in [109]. The suitability of a number of classification methods is investigated for the purpose of view independent recognition. The image datasets utilized to test the above algorithms have simple and relatively similar backgrounds, and these works didn't address the issues with complex backgrounds.

An algorithm for the recognition of hand postures in complex natural environments is useful for the real-world applications of interactive systems. The earlier works on hand posture recognition against complex backgrounds are [4, 97, 98]. Triesch *et al.* [97, 98] addressed the complex background problem using elastic graph matching. *Bunch graph method* [98] is utilized to improve the performance in complex environments. In graph algorithms, the entire image is scanned to detect the object, which increases the computational burden. In addition, in a bunch graph each node is represented using a bunch of identical node features which further decreases the processing speed. The algorithm presented in [4] does the segmentation based on skin color, and it needs fairly accurate estimates of the center and the size of the hand. The above algorithms cannot deal with complex backgrounds which contain skin colored regions, and large variations in the hand size.

This chapter focuses on the recognition of static hand gestures with

variations in hand size and with complex backgrounds. The proposed algorithm utilizes a biologically inspired approach, which is based on the computational model of visual cortex [85] and the Bayesian model of visual attention [11]. It utilizes an appearance based approach and recognizes hand postures through learning. The Bayesian theory of attention is utilized to detect and identify the hand region in the complex background hand posture images. The *where* information is extracted using feature based visual attention. Shape, texture and color features are utilized for this. The shape and texture based features are extracted from a map which represents the similarity of pixels to human skin color, using the computational model of the ventral stream of visual cortex [85]. The color features are extracted by the discretization of chrominance color components in HSI and YCbCr color spaces, and the similarity to skin color map. A saliency map is created by calculating the posterior probabilities of pixel locations to be part of a hand region, using the Bayesian model of attention. The presence of hand is detected by thresholding the saliency map, and the hand region is extracted by the segmentation of input image using the thresholded saliency map. The hand postures are recognized using shape and texture based features (of the hand region), with a Support Vector Machines (SVM) classifier.

The proposed algorithm is reliable against skin colored regions in the background as the segmentation of hand region is done using the attention mechanism, which utilizes a combination of color, shape, and texture features. The experimental results show that the proposed algorithm has a person independent performance. The proposed algorithm has robustness against variations in hand size and its position in the image.

The number of hand posture databases available to the research community is limited [95]. This thesis contributes a new 10 class complex background hand posture dataset, namely NUS hand posture dataset-II (available online [1]). The postures are obtained from 40 subjects, with various hand sizes, and with different ethnicities. The images have a variety of indoor as well as outdoor complex backgrounds. The hand postures have wide intra class variations in hand sizes and appearances. The database also contains a set of background images which is used to test the hand detection capability of the proposed algorithm. The recognition algorithm is tested with the new dataset using a 10 fold cross validation strategy. It provided an accuracy of 94.36%.

6.1 The Feature Extraction System and the Model of Attention

The biologically inspired, shape and texture based feature extraction system, and the Bayesian model of visual attention [11] are briefed in this section. The shape and texture based features are extracted using a cortex like mechanism [85], and the visual attention is implemented using these features and a set of color features.

6.1.1 Extraction of Shape and Texture based Features

The feature extraction system consists of four layers (Table 6.1). Layer 1 (S_1) consists of a battery of Gabor filters with 4 orientations (0°, 45°, 90°, 135°) and 16 sizes (divided into 8 bands). The S_1 layer imitates the

Table 6.1: Different layers in the shape and texture feature extraction system

Layer	Process	Represents
S_1	Gabor filtering	simple cells in the primary visual cortex (V1)
C_1	Local pooling	complex cells in the primary visual cortex (V1)
S_2	Radial basis functions	visual area V4 & posterior inferotemporal cortex
C_2	Global pooling	inferotemporal cortex

Note: S stands for simple cells and C stands for complex cells. The simple and complex cells are the two types of cells in the visual cortex. The simple cells primarily respond to oriented edges and bars. The complex cells provide spatial invariance.

simple cells in the primary visual cortex (V1[1]), detecting edges and bars. Layer 2 (C_1) models the complex cells in V1, by applying a MAX operator locally (over different scales and positions) to the first layer's outputs.[2] This operation provides tolerance to different object projection sizes, positions, and rotations in the 2-D plane of the visual field. In layer 3 (S_2), radial basis functions (RBFs) are utilized to imitate the visual area V4 and posterior inferotemporal (PIT) cortex. Layer 3 aids shape and texture recognition by comparing the C_1 images with prototypical C_1 image patches. The prototypical C_1 image patches (the prototype patches) are learned and stored during the training (in humans, these patches correspond to learned patterns of previously seen visual images and are stored in the synaptic weights of the neural cells). Finally, the fourth layer (C_2) applies a MAX operator (over all scales, but not over positions) to the

[1]V1, V2, V3, V4, and V5 are the visual areas in the visual cortex. V1 is the primary visual cortex. V2 to V5 are the secondary visual areas, and are collectively termed as the extrastriate visual cortex.

[2]Refer [85] for further explanation of S_1 and C_1 stages (layer 1 and 2).

outputs of layer S_2, resulting in a representation that expresses the similarities with the prototype patches. The outputs of layer 4 are C_2 response matrices, which are the shape and texture based features utilized in the attention model.

Fig. 6.1 shows an overview of the shape and texture based feature extraction system. Simple cells in the RBF stage (third layer, S_2) combines bars and edges in the image to more complex shapes. RBFs are a major class of neural network model, comparing the distance between an input signal and a prototype signal [5]. Each S_2 unit response depends in a Gaussian-like manner on the Euclidean distance between crops of the C_1 image (X_i) and the stored prototype patch (P_j). The prototype patches (*centers* of the RBF units) of different sizes are extracted from the C_1 responses of the training images. The centers of the patches are positioned at the geometrically significant and textured positions of the hand postures (Fig. 6.1). Each patch contains the four orientations. The third layer compares these patches by calculating the summed Euclidean distance between the patch (P_j) and every possible crop (X_i) of the C_1 image (combining all orientations). This comparison is done with all the C_1 responses in the second layer (the C_1 responses at different scales).

The scale invariant C_2 responses are computed by taking a global maximum over all the scales (but not over positions) for each S_2 type. Each C_2 response matrix corresponds to a particular prototype patch with a specific patch size. The more the number of extracted features the better is the classification accuracy. However, the computational burden (for feature extraction as well as classification) increases with the number of features. In the present work 15 prototype patches with 4 patch sizes are

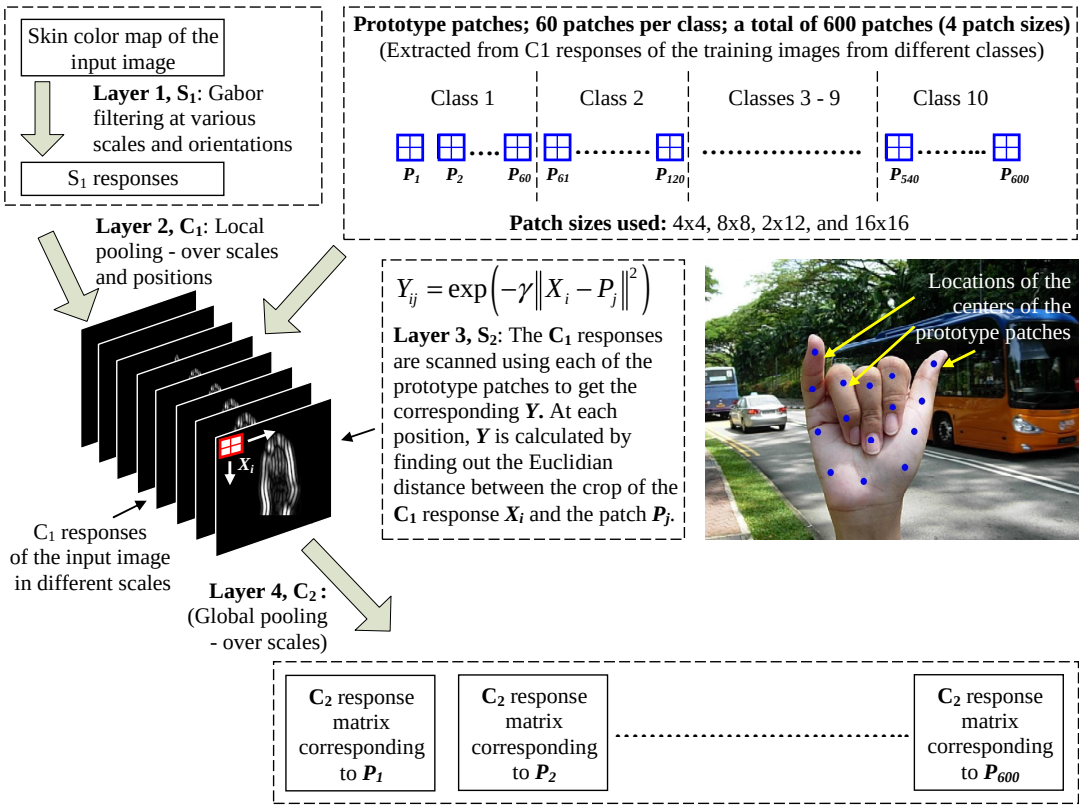


Figure 6.1: Extraction of the shape and texture based features (C_2 response matrices). The S_1 and C_1 responses are generated from a skin color map (Section 6.2.1) of the input image. The prototype patches of different sizes are extracted from the C_1 responses of the training images. 15 patches, each with four patch sizes, are extracted from each of the 10 classes leading to a total of 600 prototype patches. The centers of the patches are positioned at the geometrically significant and textured positions of the hand postures (as shown in the sample hand posture). There are 600 C_2 response matrices, one corresponding to each prototype patch. Each C_2 response depends in a Gaussian-like manner on the Euclidean distance between crops of the C_1 response of the input image and the corresponding prototype patch.

extracted from each of the 10 classes. The total number of patches is 600, leading to 600 shape and texture based features.

Modifications in the Shape and Texture Feature Extraction System

The major differences between the proposed shape and texture features (utilized in the attention system) and that presented in [85] are summarized as follows.

1. The shape and texture features are extracted from a similarity to skin color map (Section 6.2.1) (not from the grey scale image).

2. The prototype patches are extracted from different class images. The centers of the patches are placed at geometrically significant and textured positions of the hand postures.

3. The output feature component is a C_2 response matrix (instead of a real number) which retains the hand location information.

6.1.2 The Bayesian Model of Visual Attention

Visual attention is the part of the inference process that addresses the visual recognition problem of *what is where* [11]. Visual attention helps to infer the identity and position of objects in a visual scene. Attention reduces the *size* of the search space and the *computational complexity* of recognition.

The visual attention is directed selectively to objects in a scene using both bottom-up, image-based saliency cues and top-down, task-dependent

cues. The top-down task based attention is more deliberate and powerful [32], and is depended on the features of the object. The proposed pattern recognition system is task based and it utilizes a top-down approach. The attention is focussed on the region of interest using the object features.

Visual perception is interpreted as a Bayesian inference process whereby priors (top-down) help to disambiguate noisy sensory input signals (bottom-up) [16]. Visual recognition corresponds to estimating posterior probabilities of visual features for specific object categories, and their locations in an image. The posterior probabilities of location variables serve as a *saliency map*.

There are two types of visual attention, spatial attention and feature attention (Fig. 6.2). Spatial and feature based attention are utilized to reduce the uncertainty in shape and spatial information respectively. In spatial attention, different spatial priors are assigned by learning the context from the training images (the information about the position of object in the image). In feature attention, different feature priors are assigned by counting the frequency of occurrence of each feature among the training images.

A Bayesian model of spatial attention is proposed in [80]. Chikkerur *et al.* [11] modified the model to include feature based attention, in addition to the spatial attention. The model imitates the interactions between the parietal and ventral streams of visual cortex, using a Bayesian network (Bayes net). A Bayes net represents joint probability distributions in a compact manner, via conditional independence. It helps in calculating the posterior probability of an event, the probability of the event given

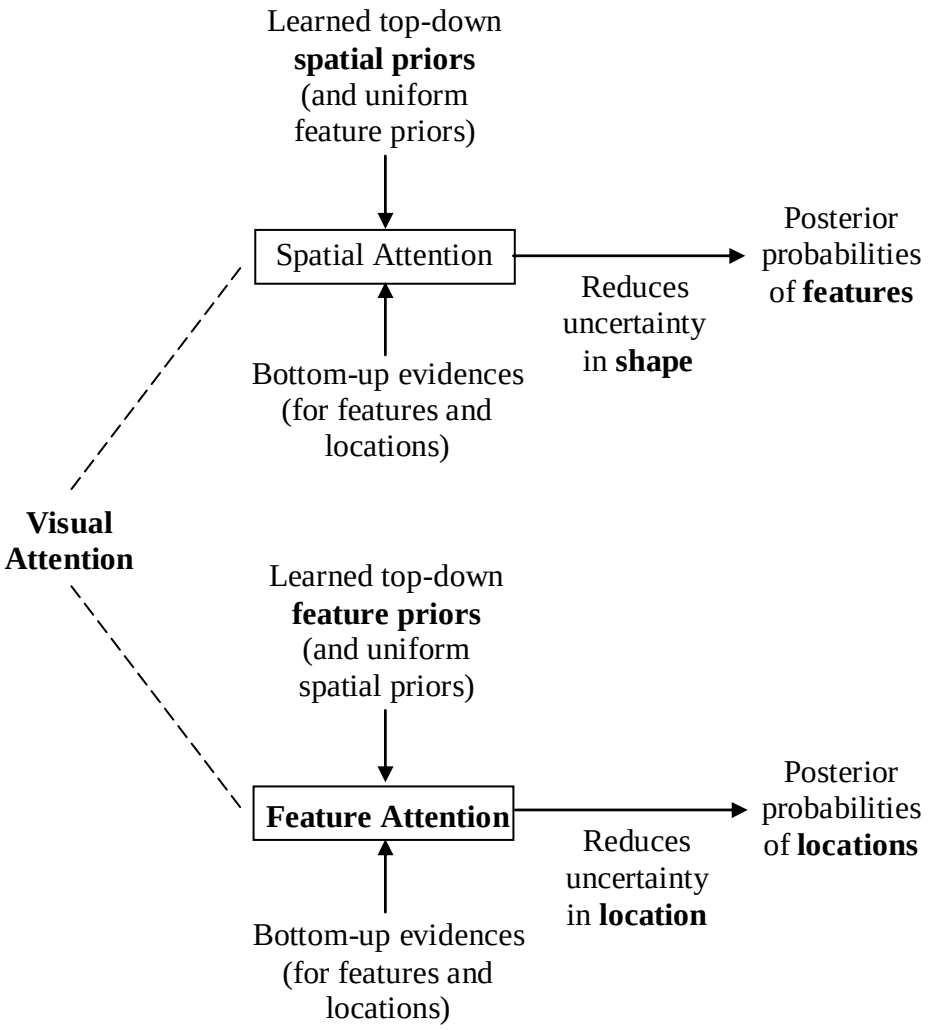


Figure 6.2: Two types of visual attention as per the Bayesian model [11, 80]. Spatial attention utilizes different priors for locations and helps to focus attention on the location of interest. Spatial attention reduces uncertainty in shape. Feature attention utilizes different priors for features and helps to focus attention on the features of interest. Feature attention reduces uncertainty in location. The output of the feature detector (with location information) serve as the bottom-up evidence in both spatial and feature attention. Feature attention with uniform location priors is utilized in the proposed hand posture recognition system, as the hand position is random in the image

any evidence, through Bayesian inference. A biologically plausible belief propagation algorithm [68] is utilized for the calculation of posterior probabilities.

The present work utilizes the feature based attention (with different feature priors) to create the saliency map. The location priors are set to be uniform, as the hand can be randomly positioned in the image. The visual attention model is developed utilizing the shape and texture based features, and the color features. The saliency map (the posterior probabilities of pixel locations) is generated using the learned feature priors and evidences from the images. The hand region is segmented by thresholding the saliency map. The complex backgrounds of the images considered contain skin colored pixels. Due to this, the utilization of color based features alone is not effective. However, when the color features are combined with shape and texture features, it resulted in better identification of the hand region, than that achieved with shape and texture features alone.

6.2 Attention Based Segmentation and Recognition

The proposed algorithm addresses the complex image background issue by utilizing a combination of different features. The hand region in the image is identified by calculating the joint posterior probability of the presence of the combination of features. Bayesian inference is utilized to create a saliency map, which helps in the segmentation of the hand region.

In Bayesian inference, the likelihood of a particular state of the world being true is calculated based on the present input and the prior knowledge about the world. The significance of an input is decided based on the prior experience. In images with complex backgrounds, the shape patterns emerging from the background affect the pattern recognition task negatively. To recognize a pattern from a complex background image, the features corresponding to the foreground object are given higher weightage compared to that corresponding to the background.

In this work, the shape and texture features of the hand postures, and the color features of the human skin are utilized to focus attention on the hand region. The posterior probability of a pixel location to be part of a hand region is calculated by assigning higher priors (which is learned from the training images) to the features corresponding to the hand area. The hand postures are classified using the shape and texture features of the hand region, with an SVM classifier.

Fig. 6.3 shows the block diagram of the proposed system. The functions of different blocks in the system are elaborated in the following subsections.

6.2.1 Image Pre-processing

The image pre-processing includes color space conversions and the generation of similarity to skin color map.

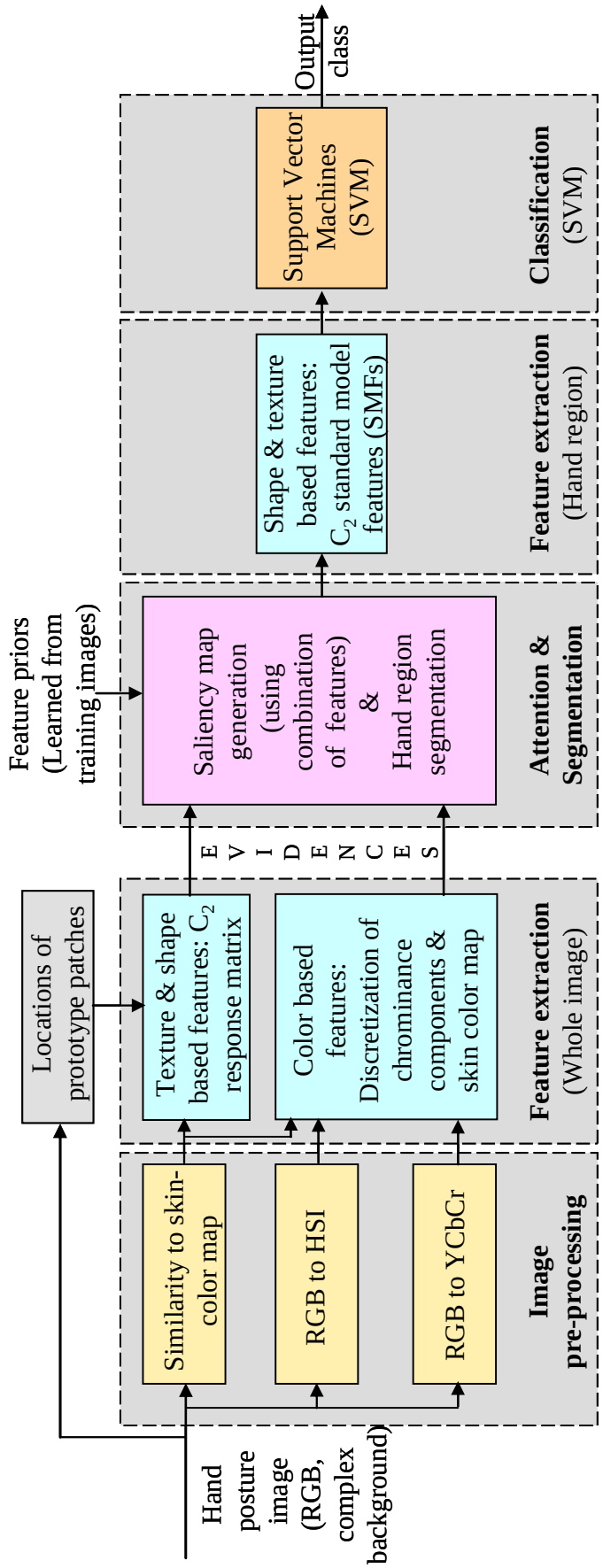


Figure 6.3: The proposed attention based hand posture recognition system.

Color Space Conversions - RGB to HSI and YCbCr

The input image in RGB space is converted to HSI and YCbCr color spaces. The conversion between RGB and HSI is nonlinear whereas that between RGB and YCbCr is linear. The chrominance components in these color spaces (H, S, C_b, and C_r) are utilized to detect the hand region in images. The hue value H refers to the color type (such as red, blue, or yellow), and the saturation value S refers to the vibrancy or purity of the color. The values of C_b and C_r represent the blue component $(B - Y)$ and the red component $(R - Y)$ [8] respectively (Y stands for the luminance value). The values of H and S are in the range [0 1], and those of C_b and C_r are within [16 240].

Similarity to Skin Color Map

A skin color map (6.1) is created using the similarity of each pixel in HSI space to the average pixel values of the skin color.

$$S_{skin} = 1 - \sqrt{(H - H_{s0})^2 + \left(\frac{H_{smax} - H_{smin}}{S_{smax} - S_{smin}}\right)^2 (S - S_{s0})^2} \qquad (6.1)$$

where,

S_{skin}	the similarity of the pixel to skin color,
H & S	the hue and saturation values of the pixel,
H_{s0} & S_{s0}	the average hue and saturation values of the skin colors,
H_{smax} & H_{smin}	the maximum and minimum of the hue values of the skin colors, and,
S_{smax} & S_{smin}	the maximum and minimum of the saturation values of the skin colors.

The average hue and saturation values are calculated by considering 10 skin colored pixel values of all the subjects.[3] The values of different parameters in (6.1), obtained from the present study, are provided in Table 6.2. The hue value span (0.1770) is smaller than that of saturation (0.5692). The coefficient of the saturation term ($(S - S_{s0})^2$) in (6.1) is a scaling factor to compensate for such a variation in span.

Table 6.2: Skin color parameters

H_{s0}	S_{s0}	H_{smax}	H_{smin}	S_{smax}	S_{smin}	Hue span ($H_{smax} - H_{smin}$)	Saturation span ($S_{smax} - S_{smin}$)
0.1073	0.3515	0.1892	0.0122	0.6250	0.0558	0.1770	0.5692

The similarity to skin color map enhances the edges and shapes within the skin colored regions in the images, while preserving the textures (Fig. 6.4). The proposed system extracts the shape and texture based features of hand postures from the skin color map. The feature extraction system detects and learns the edges and bars (at different orientations), and the textures in images. The utilization of the skin color map enhances the capability of the system to detect the hand region in complex background images.

6.2.2 Extraction of Color, Shape and Texture Features

The proposed algorithm utilizes a combination of low and high level image features to develop the visual attention model. A saliency map is generated and the attention is focussed on the hand region using color

[3]The dataset consists of hand postures by 40 subjects, with different ethnic origins, against complex backgrounds.

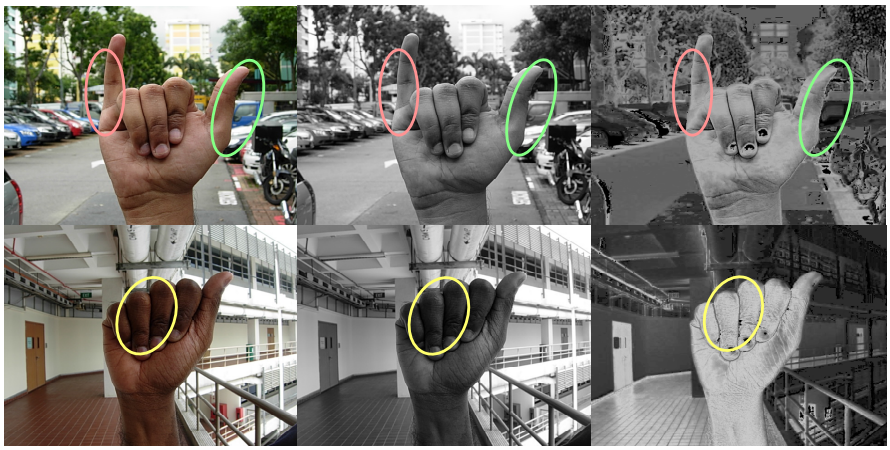


Figure 6.4: Sample hand posture images (column 1 - RGB, column 2 - grayscale) with corresponding skin color map (column 3). The skin color map enhanced the edges and shapes of the hand postures. The marked regions in column 3 have better edges of the hand, as compared with that within the corresponding regions in column 1 and 2. The edges and bars of the non-skin colored areas are diminished in the skin color map (column 3). However the edges corresponding to the skin colored non-hand region are also enhanced (row 2, column 3). The proposed algorithm utilizes the shape and texture patterns of the hand region (in addition to the color features) to address this issue.

features (low level) and shape-texture features (high level). The postures are classified using the high level features.

Color based Features

The hue (H) and saturation (S) components in the HSI color space, and, the C_b and C_r components in the YCbCr color space help to distinguish skin colored region in images. Analysis and comparison of the usage of different color spaces for skin segmentation are provided in [70], [38] and [8]. [70] rates YCbCr as the best choice for skin segmentation, whereas HSI color space is rated as the best in [8]. The proposed algorithm utilizes a combination of HSI and YCbCr color spaces. The hand region is detected using the chrominance color components (H, S, C_b, and C_r) of these color

spaces.

The proposed algorithm generated the skin color map using the average skin color components (H_{s0} and S_{s0} in Table 6.2). However the skin colors and the corresponding component values have variations about the mean value. Fig. 6.5 shows four skin samples which have inter and intra ethnic variations in skin colors. The average chrominance component values of these samples are provided in Table 6.3. The H, C_b, and C_r components have approximately 10% variation whereas that for S component is 50%. In order to detect different skin colors in spite of the variations, the proposed algorithm considers different ranges of color components.

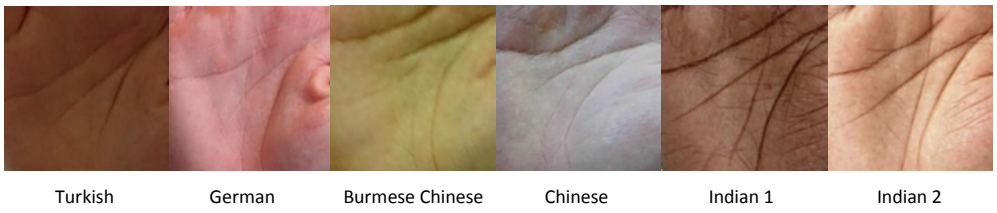


Figure 6.5: Skin samples showing the inter and intra ethnic variations in skin color. Table 6.3 provides the average H, S, C_b, and C_r values of the six skin samples.

The proposed algorithm utilizes the discretized values of chrominance color components, and the similarity to skin color map to calculate the joint posterior probability of pixel locations to be part of a hand region. The values of H, S, C_b, C_r, and S_{skin} fall within the range [0 1] (values of C_b and C_r are normalized). This range is subdivided into ten subranges in steps of 0.1. The features are named as in Table 6.4. For example the hue values between 0 to 0.1 is a feature and is named as H_1. Similarly there are 50 number of color features (Table 6.4). The prior probabilities for

the presence of these features are calculated by counting the frequency of occurrence of the features in the skin colored hand area in training images[4]. The features those had maximum priors are H_1, H_2, S_4, C_{b5}, C_{r6}, and S_{skin10}. The color features are common for all the hand postures (the position and frequency of the features may vary, however). Due to this, the color features are utilized only for focussing the attention on the hand, and not for the interclass discrimination of hand postures.

Table 6.3: Average H, S, C_b, and C_r values of the four skin samples in Fig. 6.5

Sample	H	S	C_b	C_r
Turkish	**0.1859**	0.4118	111.6	150.0
German	0.1207	0.5733	118.1	**156.4**
Burmese Chinese	0.1106	0.4529	**101.9**	141.6
Chinese	0.1830	**0.0954**	**127.8**	**129.8**
Indian 1	**0.0493**	**0.5894**	112.8	148.0
Indian 2	0.0572	0.3816	107.7	152.4

Note: The bolded figures represent the maximum and minimum values in each column.

Table 6.4: Discretization of color features

(a) Chrominance color components discretization

H	S	C_b	C_r	Ranges
H_1	S_1	C_{b1}	C_{r1}	0.0 - 0.1
H_2	S_2	C_{b2}	C_{r2}	0.1 - 0.2
H_3	S_3	C_{b3}	C_{r3}	0.2 - 0.3
H_4	S_4	C_{b4}	C_{r4}	0.3 - 0.4
H_5	S_5	C_{b5}	C_{r5}	0.4 - 0.5
H_6	S_6	C_{b6}	C_{r6}	0.5 - 0.6
H_7	S_7	C_{b7}	C_{r7}	0.6 - 0.7
H_8	S_8	C_{b8}	C_{r8}	0.7 - 0.8
H_9	S_9	C_{b9}	C_{r9}	0.8 - 0.9
H_{10}	S_{10}	C_{b10}	C_{r10}	0.9 - 1.0

(b) Skin color similarity (6.1) discretization

S_{skin}	Ranges
S_{skin1}	0.0 - 0.1
S_{skin2}	0.1 - 0.2
S_{skin3}	0.2 - 0.3
S_{skin4}	0.3 - 0.4
S_{skin5}	0.4 - 0.5
S_{skin6}	0.5 - 0.6
S_{skin7}	0.6 - 0.7
S_{skin8}	0.7 - 0.8
S_{skin9}	0.8 - 0.9
S_{skin10}	0.9 - 1.0

[4]400 images (1 image per class per subject) are considered. During the training phase the hand area is selected manually.

Shape and Texture based Features

The shape and texture descriptors (Section 6.1.1) are extracted from the similarity to skin color map. In the proposed algorithm, 15 prototype patches with 4 patch sizes, are extracted from each of the 10 classes. The 15 patches are from 15 different images of the same class (we have also experimented using patches extracted from a single image per class, but better accuracy is achieved in the earlier case as it provides better invariance). Each of the extracted patch contains the four orientations. The total number of patches is 600, leading to 600 shape and texture based features.

6.2.3 Feature based Visual Attention and Saliency Map Generation

The feature based visual attention is implemented utilizing a combination of low level (color) and high level (shape and texture) features. Fig. 6.6 shows the Bayes net utilized in the proposed system, which is developed based on the model proposed in [11]. The Bayes Net Toolbox (BNT) [58] is utilized to implement the Bayes net. The probabilistic model is given by,

$$\begin{aligned} & P(O, L, {X_s}^1, \ldots, {X_s}^{N1}, {X_c}^1, \ldots, {X_c}^{N2}, {F_s}^1, \ldots, {F_s}^{N1}, {F_c}^1, \ldots, {F_c}^{N2}, I) \\ & = P(O) \ * \ P(L) \ * \ \textstyle\prod_{i=1}^{i=N1} \left\{ P({X_s}^i / L, {F_s}^i) P({F_s}^i / O) \right\} \\ & \qquad * \ \textstyle\prod_{j=1}^{j=N2} \left\{ P({X_c}^j / L, {F_c}^j) P({F_c}^j / O) \right\} \\ & \qquad * \ P(I / {X_s}^1, \ldots, {X_s}^{N1}) \ * \ P(I / {X_c}^1, \ldots, {X_c}^{N2}) \end{aligned} \tag{6.2}$$

The feature-based attention depends on the task-based priors and evidences. The posterior probabilities of locations (6.3), which serve as a

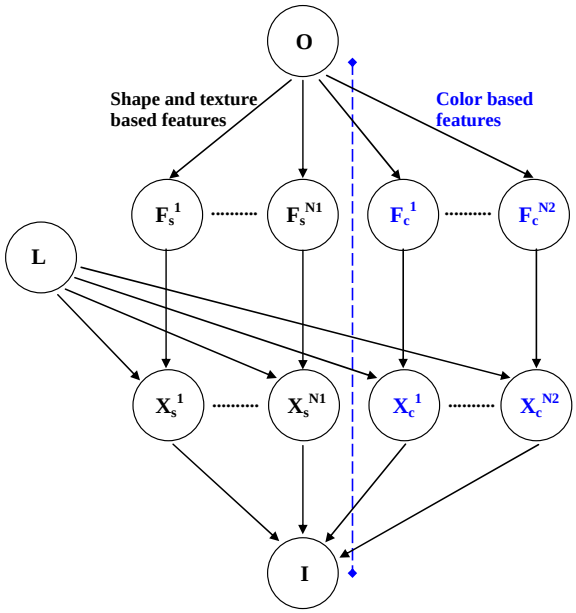


Figure 6.6: Bayes net used in the proposed system. O - the object (hand), L - the location of the hand, I - the image, ${F_s}^1$ to ${F_s}^{N1}$ - N1 binary random variables that represent the presence or absence of shape and texture features, ${F_c}^1$ to ${F_c}^{N2}$ - N2 binary random variables that represent the presence or absence of color features, ${X_s}^1$ to ${X_s}^{N1}$ - the position of N1 shape and texture based features, ${X_c}^1$ to ${X_c}^{N2}$ - the position of N2 color based features.

saliency map, are calculated using the top-down priors and bottom-up evidences. The priors and evidences are calculated from the training and testing images respectively (Table 6.5). A belief propagation algorithm [68] is utilized for the calculation of the posterior probabilities.

$$P(L/I) \propto P(L) * \prod_i \left\{ \sum_{{F_s}^i {X_s}^i} P({X_s}^i/L, {F_s}^i) P({F_s}^i/O) P(I/{X_s}^i) \right\} * \prod_j \left\{ \sum_{{F_c}^j {X_c}^j} P({X_c}^j/L, {F_c}^j) P({F_c}^j/O) P(I/{X_c}^j) \right\} \quad (6.3)$$

Table 6.5: Description of the conditional probabilities (priors, evidences, and the posterior probability)

Conditional probability	**Represents**	**Calculation**
$P(F_s^i/O)$	The top-down shape and texture feature priors; the probability of shape and texture features being present, given the presence of hand.	By counting the frequency of occurrence of features* within the training images (maximum one count per image).
$P(F_c^i/O)$	The top-down color feature priors; the probability of color features being present, given the presence of hand.	By counting the frequency of occurrence of features (Table 6.4) within the hand region in the training images (only 280 images, 1 image per class per subject, are considered).
$P(I/X_s^i)$	Bottom-up evidence for the shape and texture features; provides the likelihood that a particular location is active for the shape and texture features.	By the shape and texture feature extraction of test images (Section 6.1.1).
$P(I/X_c^i)$	Bottom-up evidence for the color features; provides the likelihood that a particular location is active for the color features.	By the color feature extraction of test images (Section 6.2.2).
$P(L/I)$	Posterior probabilities of location, which acts as the saliency map.	By the belief propagation algorithm [68].

*A feature is present if it is above a threshold value. Otherwise it is absent. In the proposed algorithm, the threshold is set at 75% of the maximum value of corresponding training features.

6.2.4 Hand Segmentation and Classification

The hand region is segmented using the saliency map generated. For the segmentation, a bounding box is created around the most salient (top 30%) locations in the image. The shape and texture based features of the hand region are extracted next. The same prototype patches selected earlier (Section 6.1.1) are utilized for the feature extraction. The C_2 SMFs are extracted by taking maximum over positions of the C_2 response matrices, similar to that done in [85]. That is, the value of the best match between a stored prototype and the input image is kept and the rest are discarded. An SVM classifier with linear kernel is utilized for the classification.

Fig. 6.7 provides an overview of the proposed system, showing the image pre-processing, feature extraction, attention, and classification stages.

6.3 Experimental Results and Discussion

The proposed algorithm is tested using a 10 class complex background hand posture dataset. Ten fold cross validation is done and the average accuracies are reported.

6.3.1 The Dataset : NUS hand posture dataset-II

As the number of available hand posture datasets is limited, a new 10 class dataset namely NUS hand posture dataset-II (Fig. 6.8, 6.9) is developed (available online [1]). The postures are shot in and around National University of Singapore (NUS), against complex natural backgrounds,

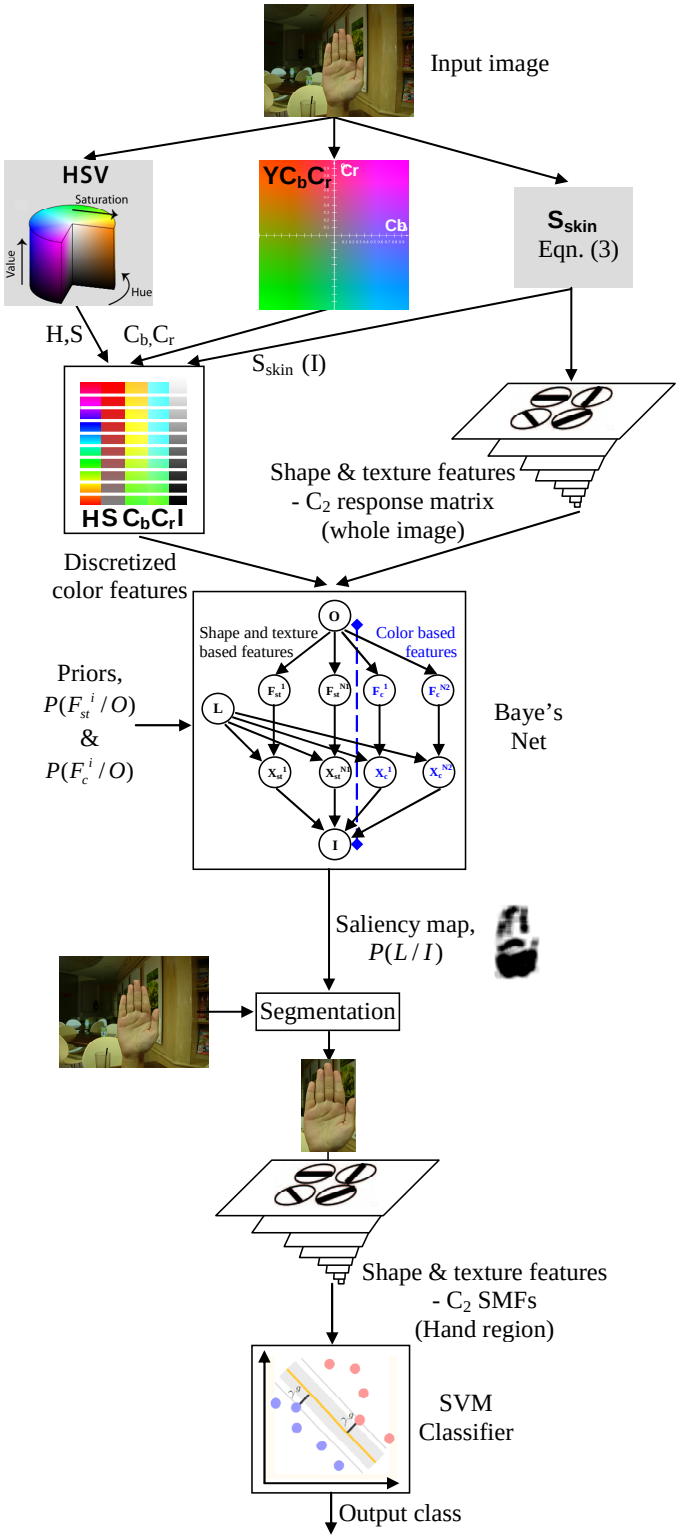


Figure 6.7: An overview of the attention based hand posture recognition system.

Figure 6.8: Sample images from NUS hand posture dataset-II, showing posture classes 1 to 10.

with various hand shapes and sizes. The dataset consists of 2000 hand posture color images with 160x120 size. The postures are obtained from 40 subjects, with various ethnicities, against different complex backgrounds. The subjects include both males and females in the age range of 22 to 56 years. The subjects are asked to show the 10 hand postures, 5 times each. They are asked to loosen the hand muscle after each shot, in order to incorporate the natural variations in the postures. The dataset also consists of 2000 background images without the hand postures.

Figure 6.9: Sample images from NUS hand posture dataset-II, showing the variations in hand postures (class 9).

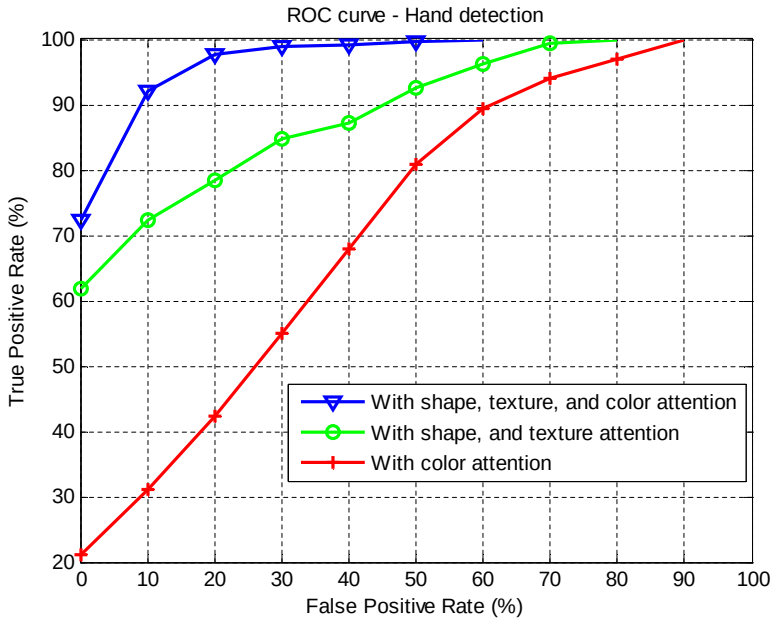


Figure 6.10: Receiver Operating Characteristics of the hand detection task. The graph is plotted by decreasing the threshold of the posterior probabilities of locations to be a hand region. Utilization of only shape-texture features provided reasonable detection performance (green) whereas utilization of only color features lead to poor performance (red) (due to the presence of skin colored backgrounds). However the algorithm provided the best performance (blue) when the color features are combined with shape-texture features

6.3.2 Hand Posture Detection

The hand postures are detected by thresholding the saliency map. To calculate the detection accuracy, saliency map is created using the posterior probabilities of locations, for the set of hand posture and the background images. If the posterior probability is above a threshold value, the presence of hand is detected. Fig. 6.10 shows the Receiver Operating Characteristics (ROC) of the hand detection task (the curve is plotted by decreasing the threshold) by the three systems; a) system with shape, texture, and color attention, b) system with shape and texture attention alone, and, c) the system with color attention alone. On comparison, the

system with shape, texture, and color attention provided better performance.

6.3.3 Hand Region Segmentation

Fig. 6.11 shows the segmentation of hand region using the skin color similarity and the saliency map. The segmentation using skin color similarity performed well when the background does not contain skin colored regions (Fig. 6.11, column 1). However natural scenes may contain many skin colored objects (more than 70% of the images in the dataset under consideration have skin colored regions in the background). The segmentation using skin color similarity fails in such cases (Fig. 6.11, column 2 and 3). The proposed attention based system succeeded in the segmentation of images with complex background, irrespective of whether it contained skin colored regions or not.

Fig. 6.12 shows 50 sample images (5 from each class) from the dataset and the corresponding saliency maps. The hand regions are segmented using these saliency maps, in a way similar to that shown in Fig. 6.11.

6.3.4 Hand Posture Recognition

The proposed hand posture recognition algorithm is tested using 10 fold cross validation strategy. The recognition accuracies for the four cases; a) with shape, texture, and color based attention, b) with shape and texture based attention, c) with color based attention, and, d) without attention are reported (Table 6.6). On comparison, the best recognition rate

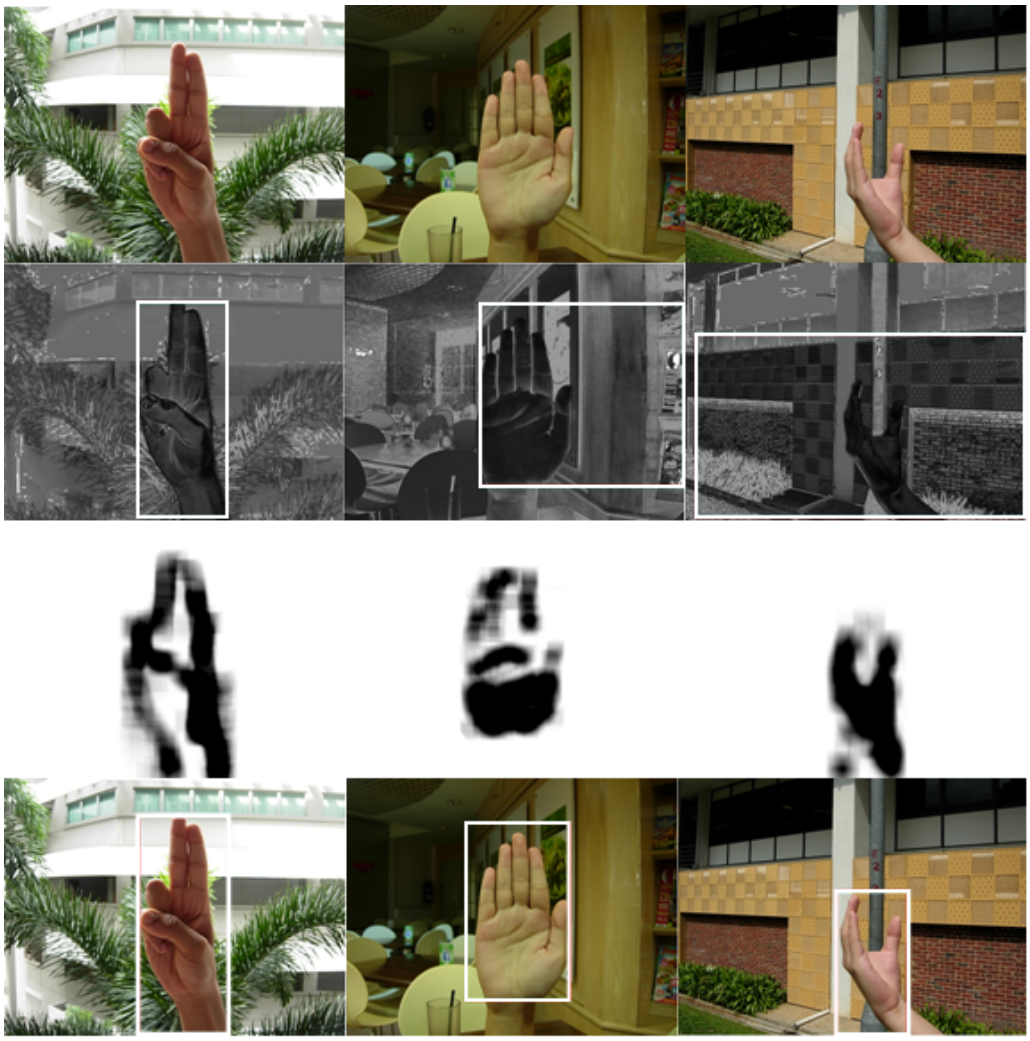


Figure 6.11: Segmentation of hand region using the similarity to skin color map and the saliency map. Each column shows the segmentation of an image. Row 1 shows the original image, row 2 shows the corresponding similarity to skin color map (darker regions represent better similarity) with segmentation by thresholding, row 3 shows the saliency map (only the top 30% is shown), and row 4 shows the segmentation using the saliency map. The background in image 1 (column 1) does not contain any skin colored area. The segmentation using skin color map succeeds for this image. Image 2 and 3 (column 2 and 3 respectively) backgrounds contain skin colored area. The skin color based segmentation partially succeeds for image 2, and it fails for image 3 (which contains more skin colored background regions, compared to that in image 2). The segmentation using the saliency map (row 4) succeeds in all the 3 cases.

Figure 6.12: Different sample images from the dataset and the corresponding saliency maps. Five sample images from each class are shown. The hand region in an image is segmented using the corresponding saliency map.

(94.36%) is achieved when the shape, texture, and color based feature attention is utilized.

Table 6.6: Hand posture recognition accuracies

Method		**Accuracy (%)**
Proposed system	Attention using shape, texture, and color features.	**94.36**
	Attention using shape and texture features alone.	87.72
	Attention using color features alone.	81.75
C_2 features without attention [85]		75.71
Elastic graph matching (EGM) [97]		69.80

When the attention is implemented using shape and texture features, the algorithm provided good improvement in the accuracy (87.72%) compared to that achieved (75.71%) by the algorithm proposed in [85]. The color feature attention alone provided lesser accuracy (80.35%) compared

to that provided by shape and texture attention. The lesser accuracy with color feature attention is due to the skin colored pixels in the background. The color features are extracted using point processing, whereas the shape-texture features are extracted using neighborhood processing. This is another reason for the lesser accuracy with color feature attention. However when color features are combined with the shape and texture features, it resulted in the best accuracy (94.36%).

Table 6.6 also shows a comparison of the accuracy provided by the proposed algorithm with that provided by the EGM algorithm [97]. The EGM algorithm provided only 69.80% recognition accuracy in spite of the high computational complexity of graph matching. The EGM algorithm performs poor when the complex background of the image contains skin colored objects. A majority of the samples misclassified by the EGM algorithm are the images with skin colored complex backgrounds. The proposed algorithm has robustness to skin colored backgrounds as it utilizes shape and texture features with color features. The shape-texture selectivity of the feature extraction system is improved as the prototype patches are extracted from the geometrically significant and textured positions of the hand postures.

6.3.5 Recognition of Hand Postures with Uniform Backgrounds

The proposed system is also tested with a simple background dataset, the NUS hand posture dataset-I [77] (Fig. 6.13). Ten fold cross validation provided an accuracy of 96.88%. This shows the effectiveness of the algorithm for recognition of postures with uniform backgrounds. However the

attention does not have much impact in the case of uniform background postures. The system without attention provided an accuracy of 95.83%, which is equivalent to that provided by the system with attention. This implies the attention based system is necessary only for the recognition in complex environments.

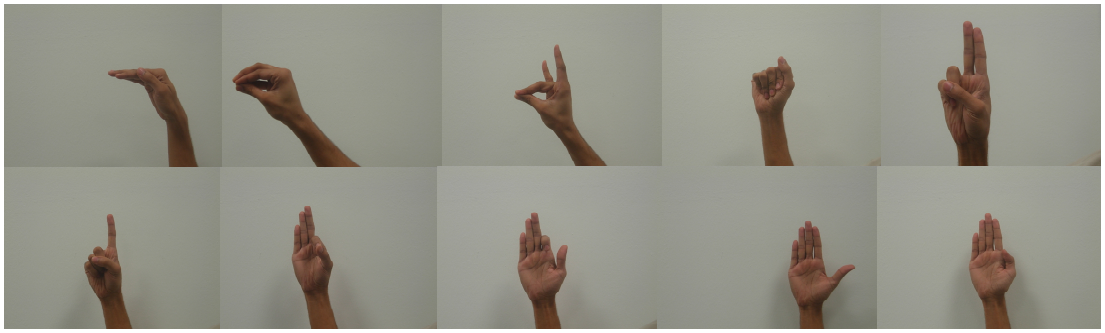


Figure 6.13: Sample images from NUS hand posture dataset-I, showing posture classes 1 to 10.

6.4 Summary

This chapter proposed an attention based system for the recognition of hand postures against complex backgrounds. A combination of high and low level image features is utilized to detect the hand, and to focus the attention on the hand region. A saliency map is generated using Bayesian inference. The postures are classified using the shape and texture based features of the hand region with an SVM classifier. The proposed algorithm is tested with a 10 class complex background dataset, the NUS hand posture dataset-II. The algorithm provided good recognition accuracy in spite of different complex backgrounds and variations in hand sizes. The algorithm is tested with color based attention alone, with

shape and texture based attention alone, and with the combination of color, shape, and texture attention. On comparison, the proposed algorithm provided the best recognition accuracy when the combination of color, shape, and texture attention is utilized.

Chapter 7

Conclusion and Future Work

This chapter provides the summary of results and contributions of the thesis. The chapter is concluded by describing the future research directions and the possible extensions of the present work.

7.1 Summary of Results and Contributions

The research reported in the thesis was motivated by the identification of unresolved issues in visual pattern recognition, in particular, the challenges in hand posture recognition. The real-time performance, variations in hand size, subject dependency, indiscernibility between similar classes, and complex backgrounds are identified as the key issues which need better solutions.

The visual images and the extracted features consist of large amount of data and information. Initial part of the thesis (Chapters 3 and 4) proposed two classification and feature selection algorithms for such multiple feature datasets. The concepts of fuzzy and rough sets are combined

to develop simple, computationally efficient, and accurate classifier algorithms. The proposed algorithms are capable of identifying the discriminative features in the dataset. Combining fuzzy and rough sets helped to classify vague and indiscernible data. The classification rules are generated by calculating the fuzzy lower and upper approximations around the data cluster centers. The proposed classifier algorithms have a polynomial time complexity. The two algorithms basically differ in the number of identified feature cluster centers. The algorithm proposed in Chapter 3 utilized only one feature cluster center (the main cluster center). The feature selection is done in an unsupervised manner, utilizing the information learned during the development of the classifier from the training dataset. This lead to a faster training phase. Two or more cluster centers are identified and utilized in the algorithm proposed in Chapter 4. More cluster centers are identified by reducing the clustering radius. This resulted in more number of classification rules and better recognition accuracy. However the unsupervised feature selection algorithm is not effective as the density of the various cluster centers (of the same feature) and the distance from the classification boundary varies independently, in a nondeterministic manner. This motivated to propose a supervised feature selection algorithm based on GA, which lead to a good *margin of classification*. The proposed feature selection and classification algorithms are applied to various image datasets (hand posture, face and object). The proposed classifier provided equivalent or better classification accuracy than that provided by an SVM classifier, with lesser computational efforts.

In order to validate the generality of the proposed feature selection and

classification algorithm for other multiple feature data, the algorithm is applied to Cancer and Tumor datasets. Five different datasets are considered. The algorithm provided classification accuracy which is equivalent or better than that reported in the literature, with less computational efforts. The maximum classification accuracy (100%) is achieved for four of the cancer / tumor datasets. The feature selection algorithm identified the predictive genes in the dataset. The proposed algorithm has the capability to classify diseases and to select relevant genes, which has high impact in the biomedical field.

The later part of the dissertation (Chapter 5) utilizes the standard model of the ventral stream of visual cortex to propose novel hand posture recognition algorithms. The effectiveness of both C_1 and C_2 features are studied. These algorithms need lesser number of training images (4 per class for the C_1 feature based algorithm, and only 1 per class for the C_2 feature based algorithm). The hand posture recognition algorithms proposed have good discriminative power between different posture classes. The C_2 features of the hand posture images are selected in such a way that it provides good interclass discrimination. The C_2 feature based recognition algorithm is implemented in real-time for the interaction of human with a virtual character. The success of the algorithm shows its utility for virtual reality applications.

A visual attention based system for the recognition of hand postures against complex backgrounds is presented in the last part (Chapter 6) of the thesis. A combination of color, texture, and shape features is utilized to improve the recognition accuracy in complex environments. The hand

region is segmented from the image using a saliency map generated using Bayesian inference. The postures are classified using the shape and texture based features of the hand region with an SVM classifier. The algorithm provided good recognition accuracy despite different complex backgrounds, and variations in hand size and appearance. The improvement in the performance of the recognition algorithm against complex backgrounds makes it suitable for real-world applications.

Two new hand posture datasets are developed for the experimental analysis of the proposed algorithms. These datasets are useful in testing and bench marking of novel pattern recognition algorithms in the research field.

7.2 Future directions

- Feature extraction: The identification of discriminative C2 features provides an insight into the functioning of the C2 feature extraction system, which imitates the primate visual cortex. There are many parameters in the system which are not tuned (and are set to match what is known about the primate visual system). Each feature corresponds to a prototype image patch extracted from the training images. The selection of relevant features identified the discriminative prototype patches, which in turn enhanced the shape selectivity. This information can be utilized to tune and select different parameters in the feature extraction system, in order to improve its performance.

- Computational complexity: The thesis proposed computationally efficient classification and feature selection algorithms for visual pattern recognition. The selection of relevant features reduced the feature extraction time as well. However the computational complexity of the biologically inspired feature extraction algorithm can be improved. In general, the computational complexity is a problem to be addressed in biologically inspired feature extraction systems.

- Dynamic gesture and Action recognition: The research proposed in the dissertation was focussed on feature extraction, selection, and classification aspects. The algorithms are applied to some of the static visual pattern classification problems. The proposed algorithms can be extended with temporal shape sequences, for the recognition of dynamic gestures and actions.

- Body posture recognition: The algorithm proposed in Chapter 6 can be extended for other shape recognition tasks like human body posture recognition, in cluttered natural environments. The utilization of color features may not be effective in the case of human body postures due to clothing on the body. However a body posture provides more reliable texture features compared to that of a hand posture. The body pose can be estimated part-by-part or hierarchically (for example, skin colored regions first and then textured regions).

Chapter 8

Author's Publications

8.1 International Journals / Conferences

1. Pramod Kumar P, Prahlad Vadakkepat, and Loh Ai Poh, "Hand posture and face recognition using a Fuzzy-Rough Approach", International Journal of Humanoid Robotics, vol.7, no.3, pp.331-356, September, 2010.

2. Pramod Kumar P, Prahlad Vadakkepat, and Loh Ai Poh, "Fuzzy-Rough discriminative feature selection and classification algorithm, with application to microarray and image datasets", Applied Soft Computing (Elsevier), vol.11, no.4, pp.3429-3440, June, 2011.

3. Pramod Kumar P, Prahlad Vadakkepat, and Loh Ai Poh, "Attention based detection and recognition of hand postures against complex natural backgrounds", International Journal of Computer Vision (under review).

4. Pramod Kumar P, Stephanie Quek Shu Hui, Prahlad Vadakkepat, and Loh Ai Poh, "Hand posture recognition using neuro-biologically inspired features", 13th Federation of International Robot soccer Association Robot World Congress, September 15-17, 2010, Bangalore,

India, TRENDS IN INTELLIGENT ROBOTICS, Springer, Book Series: Communications in Computer and Information Science, Volume: 103, Pages: 290-297, 2010.

5. Pramod Kumar P, Prahlad Vadakkepat, and Loh Ai Poh, “Graph matching based hand posture recognition using neuro-biologically inspired features”, IEEE International Conference on Control, Automation, Robotics and Vision (ICARCV 2010), Singapore, December 7-10, 2010.

6. Prahlad Vadakkepat, Pramod Kumar P, Sivakumar Ganesan, and Loh Ai Poh, “Boosting based Fuzzy-Rough pattern classifier”, 13th Federation of International Robot soccer Association Robot World Congress, September 15-17, 2010, Bangalore, India, TRENDS IN INTELLIGENT ROBOTICS, Springer, Book Series: Communications in Computer and Information Science, Volume: 103, Pages: 306-313, 2010.

Appendix A

Illustration of the formation of fuzzy membership functions, and the calculation of $\{\mu_{A_L}, \mu_{A_H}\}$ and $\{A_L, A_H\}$ - Object dataset

Fig. A.1(a)-(d) shows the identified best discriminative feature for a particular class, which has a well separated feature cluster center (the center of the fuzzy membership function). The selection of such features ease the classification process, even though there is an interclass feature overlap. Learning classification rules from the features which have distribution similar to that shown in Fig. A.2 (which have higher interclass overlap) is difficult and may lead to misclassification. The proposed algorithm (Chapter 3) neglects such features, and excludes the corresponding rules from the classifier rule base. This increases the classification accuracy, and provides better margin of classification.

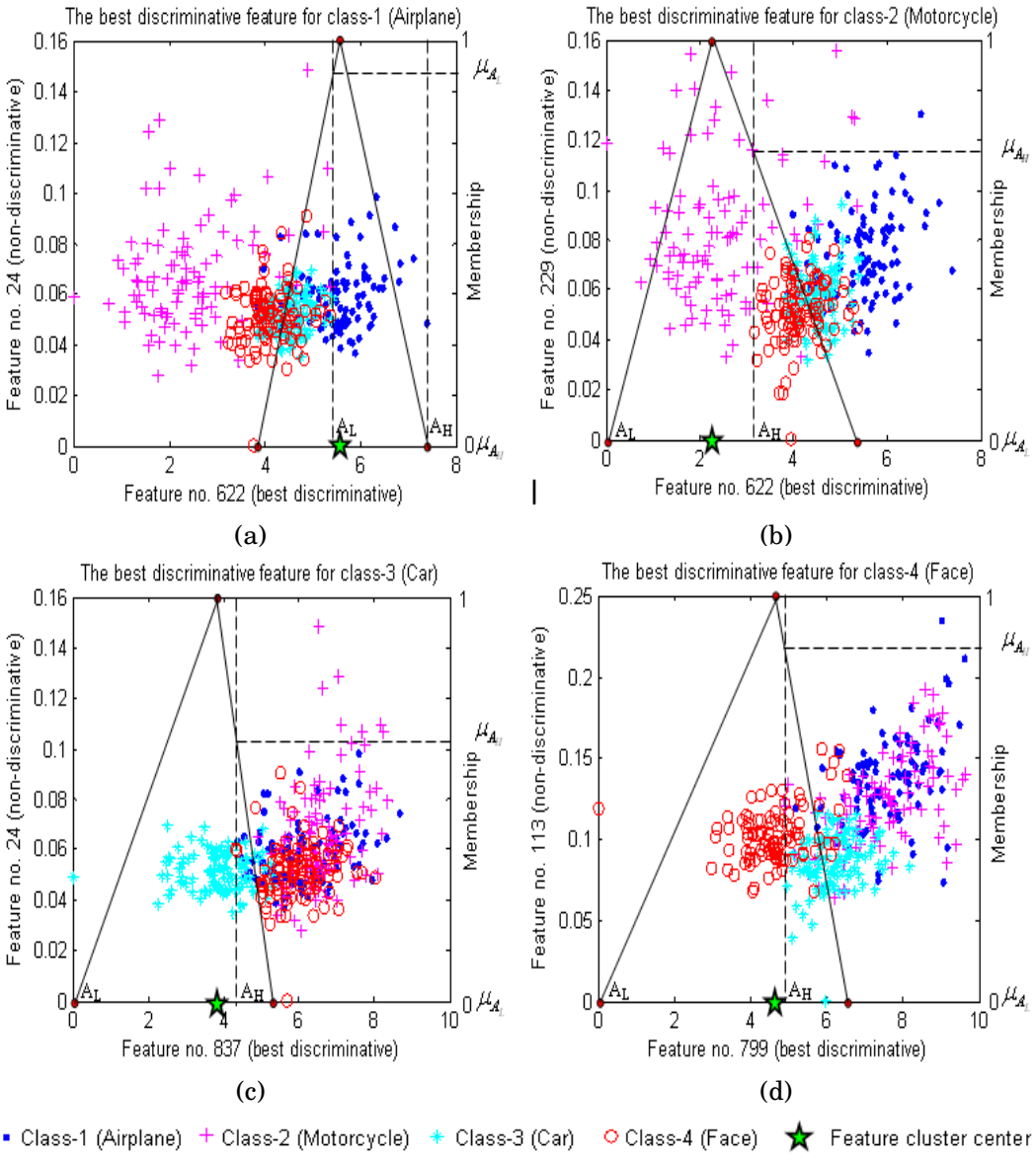


Figure A.1: Illustration of the formation of fuzzy membership functions, and the calculation of $\{\mu_{A_L}, \mu_{A_H}\}$ and $\{A_L, A_H\}$ for object dataset. Subfigures (a)-(d) show a two dimensional distribution (only two feature axes are shown) of training samples in the object dataset (class 1-4 respectively). The x-axis represents the best discriminative feature (which is selected) and the y-axis represents one of the non-discriminative features (which are not selected). Subfigures (a)-(d) also show the formation of fuzzy membership functions, the calculation of the membership values $\{\mu_{A_L}, \mu_{A_H}\}$ and the feature values $\{A_L, A_H\}$ (Section 3.2.1), for the four object classes.

Once the feature values A_L and A_H are identified, the classification is done by the voting process using *Rule 1* and *Rule 2* (4.4 and 4.5).

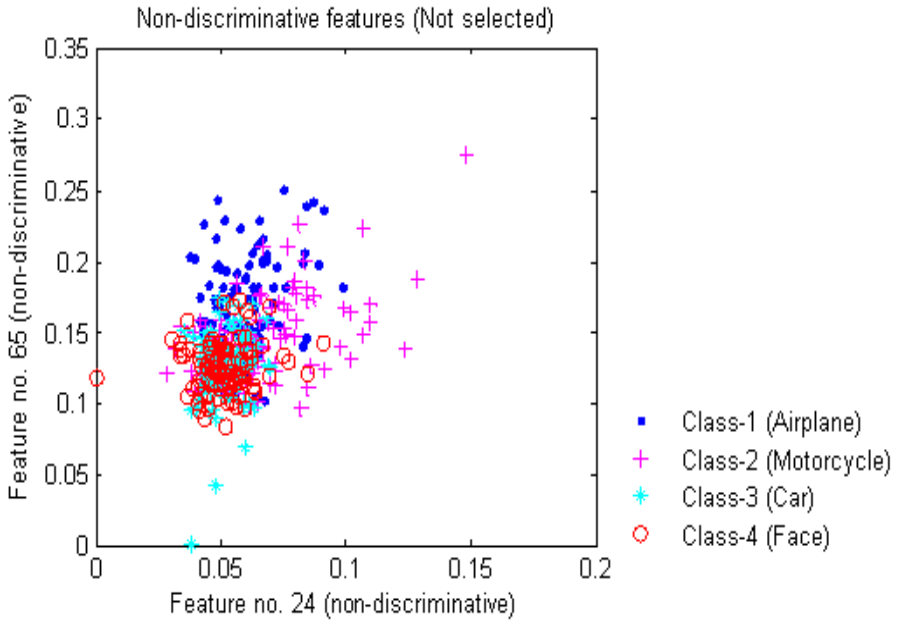


Figure A.2: Two dimensional distribution of samples in the object dataset, with x and y-axes representing two non-discriminative features. The features have high interclass overlap with the cluster centers closer to each other. Such features are discarded by the feature selection algorithm.

Bibliography

[1] *Nus hand posture dataset-ii*, 2011, http://www.ece.nus.edu.sg/stfpage/elepv/NUS-HandSet/.

[2] A. A. Albrecht, *Stochastic local search for the feature set problem,with applications to micro-array data*, Applied Mathematics and Computation **183** (2006), 1148–1164.

[3] Jonathan Alon, Vassilis Athitsos, Quan Yuan, and Stan Sclaroff, *A unified framework for gesture recognition and spatiotemporal gesture segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **31** (September, 2009), no. 09, 1685–1699.

[4] V. Athitsos and S. Sclaroff, *Estimating 3d hand pose from a cluttered image*, IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2003, pp. 432–9.

[5] C Bishop, *Neural networks for pattern recognition*, Oxford Univ. Press.

[6] F. Brill, D. Brown, and W. Martin, *Fast genetic selection of features for neural network classifiers*, IEEE Transactions on Neural Networks **3** (1992), no. 2, 324–328.

[7] C. C. Chang and C. J. Lin, *Libsvm: a library for support vector machines*, 2001, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[8] Jose M. Chaves-Gonzlez, Miguel A. Vega-Rodrgueza, Juan A. Gmez-Pulidoa, and Juan M. Snchez-Preza, *Detecting skin in face recognition systems: A colour spaces study*, Digital Signal Processing **20** (May, 2010), no. 03, 806–823.

[9] F. S. Chen, C. M. Fu, and C. L. Huang, *Hand gesture recognition using a real-time tracking method and hidden markov models*, Image and Vision Computing **21** (2003), 745–758.

[10] Q. Chen, N. D. Georganas, and E. M. Petriu, *Hand gesture recognition using haar-like features and a stochastic context-free grammar*, IEEE Transactions on Instrumentation and Measurement **57** (August, 2008), no. 8, 1562–1571.

[11] S. Chikkerur, T. Serre, C Tan, and T. Poggio, *What and where: A bayesian inference theory of attention*, Vision Research **50** (October, 2010), no. 22, 2233–2247.

[12] S. Chiu, *Fuzzy model identification based on cluster estimation*, Journal of Intelligent and Fuzzy Systems **2** (September, 1994), no. 3, 18–28.

[13] D. Conte, P. Foggia, C. Sansone, and M. Vento, *Thirty years of graph matching in pattern recognition*, International Journal of Pattern Recognition and Artificial Intelligence **18** (2004), no. 3, 265–298.

[14] K. Daniel, M. John, and M. Charles, *A person independent system for recognition of hand postures used in sign language*, Pattern Recognition Letters **31** (2010), 1359–1368.

[15] J. G. Daugman, *Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters*, J. Optical Soc. Am. A **2** (1985), no. 7, 1160–1169.

[16] P. Dayan, G. E. Hinton, and R. M. Neal, *The helmholtz machine*, Neural Computation **7** (1995), 889–904.

[17] D. Dubois and H. Prade, *Rough fuzzy sets and fuzzy rough sets*, International Journal of General Systems **17** (1990), 191–209.

[18] ———, *Putting rough sets and fuzzy sets together*, Intelligent Decision Support: Handbook of Applications and Advances in Rough Sets Theory (Roman Slowinski, ed.), Series D: System Theory, Knowledge Engineering and Problem Solving, vol. 11, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1992, pp. 203–232.

[19] Ong Eng-Jon and R. Bowden, *A boosted classifier tree for hand shape detection*, IEEE Conference on Automatic Face and Gesture Recognition, 2004, pp. 889–894.

[20] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, *Vision-based hand pose estimation: A review*, Computer Vision and Image Understanding **108** (2007), 52–73.

[21] R. Fergus, P. Perona, and A. Zisserman, *Object class recognition by unsupervised scale-invariant learning*, Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, 2003, pp. 264–271.

[22] S. S. Ge, Y. Yang, and T. H. Lee, *Hand gesture recognition and tracking based on distributed locally linear embedding*, Image and Vision Computing **26** (2008), 1607–1620.

[23] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, *From few to many: Illumination cone models for face recognition under variable lighting and pose*, IEEE Trans. Pattern Anal. Mach. Intelligence **23** (2001), no. 6, 643–660.

[24] T. R. Golub, D. K. Slonim, C.and Gaasenbeek M. Tamayo, P. Huard, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing,

M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, *Cancer program data sets*, 1999, http://www.broad.mit.edu/cgi-bin/cancer/datasets.cgi.

[25] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, *Molecular classification of cancer: Class discovery and class prediction by geneexpression monitoring*, Science **286** (1999), 531–537.

[26] M. Goodale and A. Milner, *Separate visual pathways for perception and action*, Trends in Neuroscience **15** (1992), 20–25.

[27] G. J. Gordon, R. V. Jensen, L. Hsiao, S. R. Gullans, J. E. Blumenstock, S. Ramaswamy, W. G. Richards, D. J. Sugarbaker, and R. Bueno, *Supplemental information of gordon et al. paper*, 2002, http://www.chestsurg.org/publications/2002-microarray.aspx.

[28] ———, *Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma*, Cancer Research **62** (September, 2002), 4963–4967.

[29] M. Hasanuzzamana, T. Zhanga, V. Ampornaramveth, H. Gotoda, Y. Shirai, and H. Ueno, *Adaptive visual gesture recognition for human-robot interaction using a knowledge-based software platform*, Robotics and Autonomous Systems **55**, no. 8, 643–657.

[30] X. D. Huang, Y. Ariki, and M. A. Jack, *Hidden markov models for speech recognition*, Edinburgh Univ. Press, Edinburgh, 1990.

[31] Wiesel T. N. Hubel, D. H., *Receptive fields, binocular interaction and functional architecture in the cats visual cortex*, Journal of Physiology **160** (1962), 106–154.

[32] Laurent Itti and Christof Koch, *Computational modelling of visual attention*, Nature Reviews-Neuroscience **2** (March, 2001), no. 3, 194–203.

[33] Laurent Itti, Christof Koch, and Ernst Niebur, *A model of saliency-based visual attention for rapid scene analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (November, 1998), no. 11, 1254–1259.

[34] R. Jensen and C. Cornelis, *A new approach to fuzzy-rough nearest neighbour classification*, Proceedings of the 6th International conference on Rough sets and current trends in computing, 2008, pp. 310–319.

[35] R. Jensen and Q. Shen, *Fuzzy-rough data reduction with ant colony optimization*, Fuzzy Sets and Systems **149** (2005), no. 1, 5–20.

[36] T. Jirapech-Umpai and S. Aitken, *Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes*, BMC Bioinformatics **6:148** (June, 2005).

[37] J. P. Jones and L. A. Palmer, *An evaluation of the twodimensional gabor filter model of simple receptive fields in cat striate cortex*, Journal of Neurophysiology **58** (1987), no. 6, 1233–1258.

[38] M.J. Jones and J.M.; Rehg, *Statistical color models with application to skin detection*, IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, 1999.

[39] M. Juneja, E. Walia, P. S. Sandhu, and R Mohana, *Implementation and comparative analysis of rough set, artificial neural network (ann) and fuzzy-rough classifiers for satellite image classification*, International Conference on Intelligent Agent & Multi-Agent Systems, 2009. IAMA 2009., 2009, pp. 1–6.

[40] A. Just and S. Marcel, *Interactplay dataset,two-handed datasets*, 2004, http://www.idiap.ch/resources.php.

[41] _____, *A comparative study of two state-of-the-art sequence processing techniques for hand gesture recognition*, Computer Vision and Image Understanding **113** (April, 2009), no. 4, 532–543.

[42] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P.S. Meltzer, *Microarray project*, 2001, http://research.nhgri.nih.gov/microarray/Supplement.

[43] _____, *Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks*, Nature Medicine **7** (June, 2001), no. 6, 673–679.

[44] H. D. Kim, C. H. Park, H. C. Yang, and K. B. Sim, *Genetic algorithm based feature selection method development for pattern recognition*, Proceedings of SICE-ICASE International Joint Conference 2006 (Bexco, Busan, Korea), 2006, pp. 1020–1025.

[45] M. Kolsch and M. Turk, *Robust hand detection*, IEEE Conference on Automatic Face and Gesture Recognition, 2004, pp. 614–619.

[46] B. Kwolek, *The usage of hidden markov models in a vision system of a mobile robot*, 2nd International Workshop on Robot Motion and Control (Bukowy Dworek, Poland) (K. Kozlowski, M. Galicki, and K. Tchon, eds.), pp. 257–262.

[47] M. Lades, J. C. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, R. P. Wurtz, and W. Konen, *Distortion invariant object recognition in the dynamic link architecture*, IEEE Transactions on Computers **42** (March, 1993), no. 3, 300–311.

[48] J. Lai and W. X. Wang, *Face recognition using cortex mechanism and svm*, 1st International Conference Intelligent Robotics and Applications (Wuhan, PEOPLES R CHINA) (C. Xiong, H. Liu, Y. Huang, and Y. Xiong, eds.), 2008, pp. 625–632.

[49] J. Lee and T. Kunii, *Model-based analysis of hand posture*, IEEE Comput. Graph. Appl. **15** (September, 1995), no. 5, 77–86.

[50] K. H. Lee and J. H. Kim, *An hmm based threshold model approach for gesture recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence **21** (October, 1999), no. 10, 961–973.

[51] A. Licsar and T. Sziranyi, *Dynamic training of hand gesture recognition system*, 17th International Conference on Pattern Recognition (ICPR) (Cambridge, ENGLAND) (J. Kittler, M. Petrou, and M. Nixon, eds.), pp. 971–974.

[52] ———, *User-adaptive hand gesture recognition system with interactive training*, Image and Vision Computing **23** (2005), 1102–1114.

[53] N. Liu, B. C. Lovell, and P. J. Kootsookos, *Evaluation of hmm training algorithms for letter hand gesture recognition*, 3rd IEEE International Symposium on Signal Processing and Information Technology (Darmstadt, GERMANY), 2003, pp. 648–651.

[54] J. Lu, G. Getz, E. Miska, E. Alvarez-Saavedra, J. Lamb, D. Peck, A. Sweet-Cordero, B. L. Ebert, R. H. Mak, A. A. Ferrando, J. R. Downing, T. Jacks, H. R. Horvitz, and T. R. Golub, *Micro rna expression profiles classify human cancers*, Nature **435** (June, 2005), 834–838.

[55] J. Lu, T. Zhao, and Y. Zhang, *Feature selection based-on genetic algorithm for image annotation*, Knowledge-Based Systems **21** (December, 2008), no. 8.

[56] S. Marcel, O. Bernier, J. E. Viallet, and D. Collobert, *Hand gesture recognition using input/output hidden markov models*, Proceedings of the Conference on Automatic Face and Gesture Recognition, 2000, pp. 456–461.

[57] S. Mitra and T. Acharya, *Gesture recognition : A survey*, IEEE Transactions on Systems, Man, and Cybernetics-Part C: Application and Reviews **37** (May, 2007), no. 3, 311–324.

[58] Kevin Murphy, *Bayes net toolbox for matlab*, 2003, `http://code.google.com/p/bnt/`.

[59] C. W. Ng and S. Ranganath, *Real-time gesture recognition system and application*, Image and Vision Computing **20** (2002), 993–1007.

[60] E. Niebur and C. Koch, *Computational architectures for attention*, The Attentive Brain (Cambridge, Massachusets) (R. Parasuraman, ed.), MIT Press, 1998, pp. 163–186.

[61] S. C. W. Ong and S. Ranganath, *Automatic sign language analysis: A survey and the future beyond lexical meaning*, IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (June, 2005), no. 6, 873–891.

[62] K. S. Patwardhan and S. D. Roy, *Hand gesture modelling and recognition involving changing shapes and trajectories, using a predictive eigentracker*, Pattern Recognition Letters **28** (2007), 329–334.

[63] Vladimir I. Pavlovic, Rajeev Sharma, and Thomas S. Huang, *Visual interpretation of hand gestures for human-computer interaction: A review*, IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (July 1997), no. 7, 677–694.

[64] Z. Pawlak, *Rough sets*, International Journal of Computer and Information Science **11** (1982), 341–356.

[65] ———, *Rough classification*, International Journal of Man-Machine Studies **20** (1984), 469–483.

[66] ———, *Rough sets: Theoretical aspects of reasoning about data*, Kluwer Academic Publishers, Dordrecht, 1991.

[67] ———, *Rough sets and fuzzy sets*, Proceedings of ACM, Computer Science Conference (Nashville, Tennessee), 1995, pp. 262–264.

[68] J. Pearl, *Probabilistic reasoning in intelligent systems: Networks of plausible inference.*, Morgan Kaufmann Publishers., 1988.

[69] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, *The feret database and evaluation procedure for face recognition algorithms*, Image and Vision Computing **16** (1998), no. 5, 295–306.

[70] Son Lam Phung, Abdesselam Bouzerdoum, and Douglas Chai, *Skin segmentation using color pixel classification: Analysis and comparison*, IEEE Transactions on Pattern Analysis and Machine Intelligence **27** (January, 2005), no. 01, 148–154.

[71] G. Piatetsky-Shapiro and P. Tamayo, *Microarray data mining: Facing the challenges*, SIGKDD Explorations **5** (December, 2003), no. 2, 1–5.

[72] Tomaso Poggio and Emilio Bizzi, *Generalization in vision and motor control*, Nature **431** (October, 2004), 768–774.

[73] S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel11, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub, *Prediction of central nervous system embryonal tumour outcome based on gene expression*, Letters to Nature **415** (January, 2002), 436–442.

[74] P. Pramod Kumar, Q. S. H. Stephanie, P. Vadakkepat, and A. P. Loh, *Hand posture recognition using neuro-biologically inspired features*, International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS) 2010 (Bangalore), September, 2010.

[75] P. Pramod Kumar, P. Vadakkepat, and A. P. Loh, *Graph matching based hand posture recognition using neuro-biologically inspired features*, International Conference on Control, Automation, Robotics and Vision (ICARCV) 2010 (Singapore), December, 2010.

[76] ———, *Fuzzy-rough discriminative feature selection and classification algorithm, with application to microarray and image datasets*, Applied Soft Computing **11** (June, 2011), no. 04, 3429–3440.

[77] ———, *Hand posture and face recognition using a fuzzy-rough approach*, International Journal of Humanoid Robotics **07** (September, 2010), no. 03, 331–356.

[78] H. Qinghua, A. Shuang, and Y. Daren, *Soft fuzzy rough sets for robust feature evaluation and selection*, Information Sciences **180** (November, 2010), no. 22, 4384–4400.

[79] A. Ramamoorthy, N. Vaswani, S. Chaudhury, and S. Banerjee, *Recognition of dynamic hand gestures*, Pattern Recognition **36** (2003), 2069–2081.

[80] R. Rao, *Bayesian inference and attentional modulation in the visual cortex*, Neuro Report **16** (2005), no. 16, 1843–1848.

[81] M. Riesenhuber and T. Poggio, *Hierarchical models of object recognition in cortex*, Nature Neuroscience **2** (1999), no. 11, 1019–1025.

[82] L. Rokach, *Genetic algorithm-based feature set partitioning for classification problems*, Pattern Recognition **41** (2008), 1676–1700.

[83] Amitava Roy and K. P. Sankar, *Fuzzy discretization of feature space for a rough set classifier*, Pattern Recognition Letters **24** (2003), 895–902.

[84] M. Sarkar, *Fuzzy-rough nearest neighbor algorithms in classification*, Fuzzy Sets and Systems **158** (2007), 2134–2152.

[85] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, *Robust object recognition with cortex-like mechanisms*, IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (2007), no. 3, 411–426.

[86] T. Serre, L. Wolf, and T. Poggio, *Object recognition with features inspired by visual cortex*, Conference on Computer Vision and Pattern Recognition (San Diego, CA) (C. Schmid, S. Soatto, and C. Tomasi, eds.), 2005, pp. 994–1000.

[87] Qiang Shen and Alexios Chouchoulas, *A rough-fuzzy approach for generating classification rules*, Pattern Recognition **35** (2002), 2425–2438.

[88] Christian Siagian and Laurent Itti, *Rapid biologically-inspired scene classification using features shared with visual attention*, IEEE Transactions on Pattern Analysis and Machine Intelligence **29** (February, 2007), no. 2, 300–312.

[89] W. Siedlecki and J. Sklansky, *A note on genetic algorithms for large scale feature selection*, IEEE Transactions on Computers **10** (1989), 335–347.

[90] T. Sim, S. Baker, and M. Bsat, *The cmu pose, illumination, and expression database*, IEEE Transactions on Pattern Analysis and Machine Intelligence **25** (2003), no. 12, 1615–1618.

[91] M. C. Su, *A fuzzy rule-based approach to spatio-temporal hand gesture recognition*, IEEE Transactions on Systems, Man, and

Cybernetics-Part C: Application and Reviews **30** (May, 2000), no. 2, 276–281.

[92] X. Teng, B. Wu, W. Yu, and C. Liu, *A hand gesture recognition system based on local linear embedding*, Journal of Visual Languages & Computing **16** (2005), 442–454.

[93] D. Tian, J. Keane, and X. Zeng, *Evaluating the effect of rough sets feature selection on the performance of decision trees*, Granular Computing, 2006 IEEE International Conference, 2006, pp. 57–62.

[94] J. Triesch and C. Eckes, *Object recognition with multiple feature types*, ICANN'98, Proceedings of the 8th International Conference on Artificial Neural Networks (Skovde, Swedan), 1998.

[95] J. Triesch and C. Malsburg, *Sebastien marcel hand posture and gesture datasets : Jochen triesch static hand posture database*, 1996, `http://www.idiap.ch/resource/gestures/`.

[96] ———, *A gesture interface for human-robot-interaction*, Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998 (Nara, Japan), April, 1998, pp. 546–551.

[97] ———, *A system for person-independent hand posture recognition against complex backgrounds*, IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (December, 2001), no. 12, 1449–1453.

[98] ———, *Robust classification of hand postures against complex backgrounds*, Proceedings of the Second International Conference on Automatic Face and Gesture Recognition, 1996 (Killington, VT, USA), October, 1996, pp. 170–175.

[99] Y. C. Tsai, C. H. Cheng, and J. R. Chang, *Entropy-based fuzzy rough classification approach for extracting classification rules*, Expert Systems with Applications **31** (2006), no. 2, 436–443.

[100] E. C. C. Tsang and S. Zhao, *Decision table reduction in kdd: Fuzzy rough based approach*, Transaction on Rough Sets, Lecture Notes in Computer Sciences **5946** (2010), 177–188.

[101] J. K. Tsotsos, S. M. Culhane, Y. H. Wai, W. Y. K. Lai, N. Davis, and F. Nuflo, *Modelling visual attention via selective tuning*, Artificial Intelligence **78** (October, 1995), no. 1-2, 507–545.

[102] M. Turk and A. Pentland, *Eigenfaces for recognition*, Journal of Cognitive Neuroscience **3** (1991), 71–86.

[103] E. Ueda, Y. Matsumoto, M. Imai, and T. Ogasawara, *A hand-pose estimation for vision-based human interfaces*, IEEE Transactions on Industrial Electronics **50** (August, 2003), no. 4, 676–684.

[104] T. van der Zant, L. Schomaker, and K. Haak, *Handwritten-word spotting using biologically inspired features*, IEEE Transactions on Pattern Analysis and Machine Intelligence **30** (2008), no. 11, 1945–1957.

[105] W. H. A. Wang and C. L. Tung, *Dynamic hand gesture recognition using hierarchical dynamic bayesian networks through low-level image processing*, 7th International Conference on Machine Learning and Cybernetics (Kunming, China), 2008, pp. 3247–3253.

[106] X. Wang, J. Yang, X. Teng, and N. Peng, *Fuzzy-rough set based nearest neighbor clustering classification algorithm*, Lecture Notes in Computer Science **3613/2005** (2005), 370–373.

[107] L. Wiskott, J. M. Fellous, N. Kruger, and C. Malsburg, *Face recognition by elastic bunch graph matching*, IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (July,1997), no. 7, 775–779.

[108] Y. Wu and T. S. Huang, *Vision-based gesture recognition: A review*, International Gesture Workshop on Gesture-Based Communication in Human Computer Interaction (Gif Sur Yvette, France) (A. Braffort, R. Gherbi, S. Gibet, J. Richardson, and D. Teil, eds.), Springer-Verlag Berlin, 1999, pp. 103–115.

[109] Ying Wu and T. S. Huang, *View-independent recognition of hand postures*, IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2000, pp. 88–94.

[110] F.F. Xu, D.Q. Miao, and L. Wei, *Fuzzy-rough attribute reduction via mutual information with an application to cancer classification*, Computers & Mathematics with Applications **57** (March, 2009), no. 6, 1010–1017.

[111] J. Xuan, Y. Wang, Y. Dong, Y. Feng, B. Wang, J. Khan, M. Bakay, Z. Wang, L. Pachman, S. Winokur, Y. Chen, R. Clarke, and E. Hoffman, *Gene selection for multiclass prediction by weighted fisher criterion*, EURASIP Journal on Bioinformatics and Systems Biology **2007** (2007).

[112] R. Yager and D. Filev, *Generation of fuzzy rules by mountain clustering*, Journal of Intelligent and Fuzzy Systems **2** (1994), no. 3, 209–219.

[113] H. D. Yang, A. Y. Park, and S. W. Lee, *Gesture spotting and recognition for humanrobot interaction*, IEEE Transactions on Robotics **23** (April, 2007), no. 2, 256–270.

[114] M. H. Yang and N. Ahuja, *Extraction and classification of visual motion patterns for hand gesture recognition*, Proceedings, IEEE Conference on Computer Vision and Pattern Recognition (Santa Barbara, CA, USA), 1998, pp. 892–897.

[115] M. H. Yang, N. Ahuja, and M. Tabb, *Extraction of 2d motion trajectories and its application to hand gesture recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (August, 2002), no. 8, 1061–1074.

[116] X. Yin and M. Xie, *Estimation of the fundamental matrix from uncalibrated stereo hand images for 3d hand gesture recognition*, Pattern Recognition **36** (2003), 567–584.

[117] H. S. Yoon, J. Soh, Y. J. Bae, and H. S. Yang, *Hand gesture recognition using combined features of location, angle and velocity*, Pattern Recognition **34** (2001), 1491–1501.

[118] Lotfi A. Zadeh, *Fuzzy sets*, Information and Control **8** (1965), no. 3, 338–353.

[119] M. Zhao, F. K. H. Quek, and X. Wu, *Rievl: Recursive induction learning in hand gesture recognition*, IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (November, 1998), no. 11, 1174–1185.

[120] S. Zhao, E. C. C. Tsang, D. Chen, and X. Wang, *Building a rule-based classifier-a fuzzy-rough set approach*, IEEE Transactions on Knowledge and Data Engineering **22** (2010), no. 5, 624 – 638.

[121] H. Zhou and T. S. Huang, *Tracking articulated hand motion with eigen dynamics analysis*, In proceedings of International Conference on Computer Vision, vol. 02, 2003, pp. 1102–1109.